An optimization model for multi-deep storage

Francesc Bou Manobens

Supervisor: Prof. Dr. Birger Raa

Counsellors: Geert Cosyn (Logflow), Simon Popelier (Logflow)

Master's dissertation submitted in order to obtain the academic degree of Double Master in Industrial and Management Engineering for the Polytechnic University of Catalonia, UPC.

Department of Industrial Systems Engineering and Product Design
Chair: Prof. Dr. El-Houssaine Aghezzaf
Faculty of Engineering and Architecture
Academic year 2018-2019

An optimization model for multi-deep storage

Francesc Bou Manobens

Supervisor: Prof. Dr. Birger Raa

Counsellors: Geert Cosyn (Logflow), Simon Popelier (Logflow)

Master's dissertation submitted in order to obtain the academic degree of Double Master in Industrial and Management Engineering for the Polytechnic University of Catalonia, UPC.

Department of Industrial Systems Engineering and Product Design
Chair: Prof. Dr. El-Houssaine Aghezzaf
Faculty of Engineering and Architecture
Academic year 2018-2019

GHENT
UNIVERSITY

Acknowledgments

First, I would like to thank my Master Thesis Supervisor Dr.Birger Raa, who at the time he gave me the opportunity to carry out this Master Thesis with pleasant guidance and introduced me to the research and study of logistics, particularly in the Industrial Systems Optimization and Control (ISyOC). It has been fascinating to deepen my knowledge in the warehouse elements and features and to look into the design and optimization of storying systems. Thank you for your trust.

I would also like to reflect on the support received from Geert Cosyn and Simon Popelier, logistic consultants that work in Logflow. Since the beginning they have provided me with all the required information to perform and develop the project. The first meetings were very useful to get the general framework of the Master Thesis and to find the best approach to create the optimization model for multi-deep storage systems.

Finally, I would to thank the unconditional support of my family, specially to my mother Rosa Maria, my father Josep and my brother Jordi. They are always there to support my decisions and to encourage me in my challenges.

.

28/05/2019

Abstract

Multi-deep storage systems have seen many implementations in warehouses due to their high floor space utilization. Setting the optimal lane depth for the incoming products has a considerable influence on the space utilisation and the storage efficiency, as well as the layout of the storage zones and the selection of the storage modes, the handling equipment and all the induced costs. Conventional models in designing block stacked warehouse assume uniform and deterministic inflow and outflow of products in specific quantities and time intervals. These assumptions would lead to underestimate the space required for each specific case. In this study a recursive model is developed to address the decisions of the combination of single-deep and multi-deep lanes of different depth of a warehouse when flow of products is stochastic and dynamic in nature. Furthermore, the model gives additional value to the designer to maximize warehouse space efficiency, and thus, diminishing the costs. The main objective is to find out the combination of single-deep lanes and multi-deep lanes with different depths that make up the storage system.

Keywords - SKU, single-deep, multi-deep, lane depth, honeycomb losses, accessibility losses.

AN OPTIMIZATION MODEL FOR MULTI-DEEP STORAGE

Fracesc Bou Manobens, Birger Raa
Department of Industrial Systems Engineering and Product Design
Ghent University
Technologiepark-Zwijnaarde 46, 9052 – Gent – Belgium
E-mail: francescbou4@gmail.com

ABSTRACT

Multi-deep storage systems have seen many implementations in warehouses due to their high floor space utilization. Setting the optimal lane depth for the incoming products has a considerable influence on the space utilisation and the storage efficiency, as well as the layout of the storage zones and the selection of the storage modes, the handling equipment and all the induced costs. Conventional models in designing block stacked warehouse assume uniform and deterministic inflow and outflow of products in specific quantities and time intervals. These assumptions would lead to underestimate the space required for each specific case. In this study a recursive model is developed to address the decisions of the combination of single-deep and multi-deep lanes of different depth of a warehouse when flow of products is stochastic and dynamic in nature. Furthermore, the model gives additional value to the designer to maximize warehouse space efficiency, and thus, diminishing the costs. The main objective is to find out the combination of single-deep lanes and multi-deep lanes with different depths that make up the storage system.
Keywords **- SKU, single-deep, multi-deep, lane depth, honeycomb losses, accessibility losses.**

1 INTRODUCTION

The problem considered in this paper is taken from a real-life issue. Conventional models in designing storage systems assume uniform and deterministic incoming and demand in specific quantities and time intervals. The adopted storage infrastructures significantly influence the warehouse performance in terms of space and time efficiency in addition to the initial investments costs that can differ substantially. Logflow is a consulting company that offers master warehouse operations solutions in the entire logistics, the hard automation and IT automation. In order to improve their current services in the design of suitable storage systems adapted to each client and its needs, they are looking for scientific research to build an optimization model for multi-deep storage systems. For storage of pallets, single-deep pallet racking is the most flexible solution, each pallet being individually reachable. However, it is not always the optimal solution when taking into consideration other factors as space utilization. One of the possible solutions to increase storage density are multi-deep storage systems (floor storage, drive-in racking, shuttle racking etc.).

The objective of this Master Thesis is to build a calculation and optimization model for multi-deep storage systems. The purpose is to compare multi-deep with single-deep pallet storage systems and to find out the optimal depth or combinations of depths in order to get the maximum space efficiency.

The model will take into consideration the product and environment parameters as the stock size, the reception size, the stock rotation in the warehouse, stackability (possible number of levels in the building), etc. The model has to aid of the design of new storage systems from green field determining the optimal combination of single-deep and multi-deep of different depth of a warehouse when flow of products is stochastic and dynamic in nature.

2 LITERATURE REVIEW

The determination of the optimal lane depth in multi-deep storage system has a great impact on layout problems and has been widely studied. For the space efficiency objective of this Master Thesis, the determination of the optimal lane depth for the products is the central issue, as a storage lane remains unavailable for arriving pallets until its current content has been totally depleted by demand, thereby creating the need to optimize storage lane depth (Cormier et. al 1992). With this goal, most of the following literature review relates to the maximization of the utilization of storage space and specifically to the optimal lane depth.

In 1965 the optimal lane depth for single product was determined considering the floor space utilization for the first time (Kind 1965, 1975). Then, a new model was proposed to balance the required number of lanes including the put-away and retrieving activities (Kooy 1975). This model introduced the idea that an optimal warehouse system should have an empty lane of the depth required to store an arriving product optimally available. A GPSS simulation model was created to determine if some performance indicators such as primary storage area and aisle area are affected by the effect of alternate lane depths on volume utilization (Marsh 1979). Some years later, there was improvement, and single depth for all the stored products was determined (Matson et. al 1981, 1984). Ten years later, a better procedure was developed determining the optimal number of lanes and optimal lane depths for single products following a triangular pattern and comparing the results with several heuristics (Goetschalckx et. al 1991).

The concept of *cube-utilization index* is defined as the space needed to store the products including the accessibility space (Kay et. al 2009). In this research they explored how the cube utilization aids the determination of the optimal lane depth in presence of dedicated and random storage configurations They determined that the lane depth that maximizes the cube utilization corresponds to the best compromise between honeycomb loss and down-aisle space.

All of these patterns take into consideration a constant incoming and demand with the same number of SKU per lot in all the horizon time. This can be applicable in some cases, but in the majority of cases, it is far away from the real-world warehousing systems. To face a seasonable demand, a decision-support model (ILP) was investigated, to the optimal lane depth, storage mode and zone or the assignment of the incoming products (Accorsi et al. 2016). However, there has not been research for random incoming and demand. In this Master Thesis, this issue will be dealt with, to find out the best possible combination of single-deep and multi-deep lanes for random demand and production.

3 PROBLEM STATEMENT

The inventory control system is based on the following assumptions:

- Both for incoming quantities (from production or suppliers) and outgoing quantities (to satisfy customer demands). There is not a fixed quantity per cycle time, it changes from cycle to cycle and is random data that does not follow any pattern.
- These are the main practical considerations that add a degree of complexity that the previous research had not yet considered.
- Both for incoming and outgoing orders, the entire order is received in one batch and the replenishment is instantaneous.
- The stock of day t is the is the sum of the stock of day $t-1$ and the incoming of day t minus the demand of day t.

3.1. NOTATION

I:1,..,I	[u]	*SKUs.*
K:1,..,K	[pallet]	Lane depth.
t: 1,..,T	[day]	Time t with a given horizon T.
q_{it}	[pallet]	Expected incoming of SKU i at time t (batch).
d_{it}	[pallet]	Expected demand of SKU i at time t.
s_{it}	[pallet]	Stock of SKU i at time t.
$s_{q_{it}}$	[pallet]	Stock of the incoming of SKU i at time t (q_{it}).
$s_{q_{it}sd}$	[pallet]	Stock of the incoming of SKU i at time t (q_{it}) stored in a single deep lane.
$s_{q_{it}mdk}$	[pallet]	Stock of the incoming of SKU i at time t (q_{it}) stored in a multi-deep lane of depth k.
$b_{q_{it}mdk}$	[pallet]	Stock of the incoming of SKU i at time t (q_{it}) stored in a multi-deep lane of depth k that full partially a lane.
$c_{q_{it}mdk}$	[row]	Number of rows of the incoming of SKU i at time t (q_{it}) stored in a multi-deep lane of depth k that full partially a lane.
$k_{qit\ mdk}$	[row]	Depth of the lane of the incoming of SKU i at time t (q_{it}) stored in a multi-deep lane of depth k.
$l_{qit\ sd}$	[u]	Lanes of the incoming of SKU i at time t (q_{it}) stored in a single-deep lane.
$l_{qit\ mdk}$	[u]	Lanes of the incoming of SKU i at time t (q_{it}) stored in a multi-deep lane of depth k.
l_{itsd}	[u]	Lanes of SKU i at time t of single-deep lanes.
l_{itmdk}	[u]	Lanes of SKU i at time t of multi-deep lanes of depth k.
a	[m]	Accessibility distance required by each lane.
z	pallet	Stackability levels of the storage system.
N	u	Number of different areas in the warehouse.
Subscripts		
M		1000000
max		Maximum

Table 1. Data and variables of the optimization of a multi-deep storage system.

3.2. CONSTRAINS

Below are the main boundaries and constrains of the model:

- Ensure that the multi-deep storage system has a limited number of areas with lanes of different depth.
- Guarantee that a SKU i cannot be mixed with another SKU i at the same lane.
- Ensure that a q_{it} (incoming of SKU i at day t) cannot be mixed with another q_{it} of the same SKU i at another day t.
- The model is only prepared to determine the best combination of single-deep and multi-deep-lanes for drive-in storage systems (Floor storage can be determined with this model setting a z=1); Shuttle racking cannot be determined due to the second and third constraint).
- The model is not prepared to conduct stock breakdown situations, it is necessary to prepare the incoming and demand data at the beginning to ensure that this will not happen.

4. OPTIMIZATION MODEL

We propose a recursive model that assigns the incoming products to the optimal lane depth by minimizing the space losses generated by accessibility and honey combing inefficiencies.

The suggested model follows a bottom-up strategy that is composed of three main process. The first process consists of solving the problem in a local scope, handling each product i separately and independently. The best possible number of single-deep or multi-deep lanes for each product can thus be found. The output from this local analysis is usually a solution with a large number of areas with lanes of different depths. Then, to find a more holistic and better global solution, the model proceeds to a relocation of the products in a particular number of areas with a specific depth based on the results of the previous process. Hence, to initiate the path in finding a global solution, it transfers the output from a large number of areas with lanes of different depths to a solution with a particular number of areas with a specific depth. Finally, the model focusses on determining which will be the best solution globally.

The output data of the model are:

- The **combination of single-deep and multi-deep lanes of different depth**:
- The **surface of the warehouse (m^2):** It comes from adding the space required by the storage system structure to the aisle space required for the lift trucks to manoeuvre inside the warehouse.
- The **storage capacity (pallet locations)**.
- The **storage efficiency (%)**: The fill rate relative to the gross number of pallet locations.
- The **surface efficiency (pallet/m^2):** The fill rate relative to the floor surface.
- The **cost of the warehouse (€):** It is calculated based on the cost per pallet location of the different storage systems (multi-deep or single-deep), the cost of the floor (€/m^2) that will be required to build the structure of the storage system and the cost of the handling equipment (lift truck).
- The **cost per net pallet location (€/pallet)**: It is calculated dividing the cost of the warehouse by the storage capability.

The methodology used to approach the final solution is divided in the following three main processes: The **"Local Optimization"**, the **"Relocation"** and the **"Redistribution"** process.

4.1. LOCAL OPTIMIZATION

"Local Optimization" is the first process applied to the model to get a local solution for each product in which it will be possible to know the best possible combination of single-deep and multi-deep lanes of different depth for each product i at each day t to minimize the space losses. To carry out this process the algorithm will perform the next steps:

Step 0. (Initialization) Set the number of products that have to be stored in the warehouse ($i=1,..I$), the rack dimensions (r_l, r_h y r_d), the stackability (z), the aisle width (a) and the costs variables (c_{sd}, c_{md}, c_s and c_h).

Step 1. Read the initial data of product i to determine the q_{it} (incoming of product i at day t), the q_{it} (demand of product i at day t), the horizon time ($t = 1,..,T$) and d_{qit} (consumption of the batch q_{it} at day t).

Step 2. Carry out the **Pre-Process** to calculate the maximum number of lanes of single deep (*maximum lanes sd*) and maximum depth (*maximum k-depth*).

Step 3. Initialisation of the variables of the *time*, the *stock* and the *Indicator_def*: $t = 1$; $s_{it} = 0$; $s_{md} = 0$; $s_{qitsd} = 0$; $s_{qitmd} = 0$; $Indicator\_def = M$.

Step 4. If at day t the incoming of product i is bigger than 0 ($q_{it} > 0$) execute the **"Incoming Distribution"** process.

Step 5. If at day t the demand of product i is bigger than 0 ($d_{it} > 0$) execute the **"Inventory Control"** process.

Step 6. The algorithm moves to next day. Update the *time, the stock* and the *Indicator_def*: $t = t+1$; s_{sd}; l_{sd}; s_{md}; l_{md}; s_{qitsd}; l_{qitsd}; s_{qitmdk}; l_{qitmdk}; b_{qitmdk}; c_{qitmdk}; k_{qitmdk}; $Indicator\_def = M$.

Step 7. Repeat Step 1 to Step 6 for all the products. Update i: $i = i + 1$.

Step 8. The process finishes when there are no more products to store in the warehouse ($i = I$).

Looking at the Local Optimization more thoroughly, the three main activities carried out by the algorithm are the "**Pre-process**", the **"Incoming Distribution"** and the **"Inventory Control"**. These are outlined below:

Pre-process: It calculates the *maximum number of lanes of single-deep* and the *maximum lane depth* to make sure the algorithm covers all the candidate solutions; it will set the limits of the exploration.

- **Maximum number of single-deep lanes (maximum lanes sd):**

$$s_{it} = \begin{cases} q_{it} - d_{it} & \text{if } t = 1 \\ s_{it-1} + q_{it} - d_{it} & \text{if } t = 2,..,T \end{cases} \tag{1}$$

$$\text{maximum stock level}_i = \max s_{it} \quad \forall t \tag{2}$$

$$\text{maximum lanes sd}_i = \frac{\text{maximum stock level}_i}{z} \tag{3}$$

- **Maximum lane depth (maximum k-depth):**

$$\text{maximum incoming}_i = \max q_{it} \quad \forall t \tag{4}$$

$$maximum\ k-depth = \frac{maximum\ incoming_i}{z} \tag{5}$$

Incoming Distribution: The main objective of this process is to locate the incoming product i to a single-deep lane, to a multi-deep lane or to both of them in the most efficient way in terms of floor space utilisation. To ensure the model chooses the best option it explores the five different potential options and calculates the indicator for all of the options: For all the different depths ($k = 2,..,maximum\ k\text{-}depth$) and for all the possible number of single-deep lanes ($lsd = 1,..,Maximum\ lanes\ sd$).

Step 0. (Check q_{it}): If at day t the incoming of product i is bigger than 0 ($q_{it} > 0$) execute the **"Incoming Distribution"** process.

Step 1. It starts the iterative process:

Set the variable k: $k = 2$.

Step 1.1. Update k: $k = k+1$.

Set the variable lsd: $lsd = 0$.

Step 1.2. Update lsd: $lsd = lsd +1$.

Determine the possible combinations of single-deep and multi-deep lanes and the stock distribution in the different lanes following the variables k and lsd.

Case 1 ("Prior_sd_tot1"): All the q_{it} is stored in single-deep lanes.

Calculate the Indicator (*Indicator_Prior_sd_tot1 = Indicator*).

Case 2 (***"Prior_sd_tot2")***: All the q_{it} is stored in single-deep lanes. But, if there is not enough space in the single-deep, the remaining q_{it} is located in multi-deep lanes of depth k.

Calculate the Indicator (*Indicator_Prior_sd_tot2 = Indicator*).

Case 3 ("Prior_md_tot1"): All the q_{it} is stored in multi-deep lanes of depth k.

Calculate the Indicator

(*Indicator_Prior_md_tot1 = Indicator*).

Case 4 ("prior_md_tot2"): All the q_{it} is stored in multi-deep lanes. But, if there is some pallet of the q_{it} that full partially a lane of the multi-deep lanes of depth k and is smaller than the available space of the single-deep lanes, then it is relocated to a single-deep lane.

Calculate the Indicator.

(Indicator_prior_md_tot2 = Indicator).

Case 5 ("Prior_md_tot3"): All the q_{it} is stored in multi-deep lanes of depth k. But, if there is some pallet of the q_{it} that full partially a lane of the multi-deep lane, then it is relocated to a single-deep lane (even though there could not be space, it would be necessary to add a new single-deep lane in the warehouse).

Calculate the Indicator

(*Indicator_Prior_md_tot3 = Indicator*).

Select the minimum Indicator.

(Check the Indicator) If the Indicator is smaller than the *Indicator_def* (*Indicator < Indicator_def*) set the Indicator as the new *Indicator_def* and save the inventory variables

(s_{sd}; l_{sd}; s_{md}; l_{md}; s_{qitsd}; l_{qitsd}; s_{qitmdk}; l_{qitmdk}; b_{qitmdk}; c_{qitmdk}; k_{qitmdk}).

Repeat *Step 1.2.* until there are no more possibilities to explore (*lsd = maximum lanes sd).*

Repeat *Step 1.1.* until there are not more possibilities to explore *(k = maximum k-depth).*

Inventory Control Process: The main objective of this process is to decide which pallets of product *i* should be removed, from a single-deep lane, from a multi-deep lane or from both of them. To ensure that the model chooses the best option in terms of space efficiency the model explores the different potential options and calculates the indicator for all of the options. The main rule is to follow a FIFO (First In, First Out) policy to satisfy the demand. Below is the

Step 0. (Check d_{it}): If at day *t* the incoming of product *i* is bigger than 0 ($d_{it} > 0$) execute the **"Inventory Control"** process. Set $d = d_{it}$.

Step 1. (Check s_{qitsd} and s_{qitmdk}):

Step 1.1. If there is s_{qitsd} and s_{qitmdk} available ($s_{qitsd} > 0$ and $s_{qitmdk} > 0$):

Func priority sd

If the demand is smaller than the oldest available stock of the single-deep lanes ($d < s_{qitsd}$) remove as many stock as *d* ($s_{qitsd} = s_{qitsd} - d$ and $d = 0$).

If the demand is bigger than the oldest available stock of the single-deep lanes ($d_{it} > s_{qitsd}$) remove all the stock ($s_{qitsd} = 0$ and $d = d - s_{qitsd}$).

Calculate Indicator (*Indicator_priority_sd = Indicator*).

Func priority md

If the demand is smaller than the oldest available stock of the multi-deep lanes ($d < s_{qitmdk}$) remove of as many stock as d ($s_{qitmdk} = s_{qitmdk} - d$ and $d = 0$).

If the demand is bigger than the oldest available stock of the multi-deep lanes ($d > s_{qitmdk}$) remove all the stock ($s_{qitmdk} = 0$ and $d = d - s_{qitmdk}$).

Calculate the Indicator (*Indicator_priority_md = Indicator*).

If Indicator priority sd is smaller than Indicator priority md execute **Func priority sd**, otherwise execute the **Func priority md**.

Step 1.2. If there is s_{qitsd} or s_{qitmdk} available ($s_{qitsd} > 0$ or $s_{qitmdk} > 0$):

If there is only s_{qitsd} available ($s_{qitsd} > 0$ and $s_{qitmdk} = 0$):

Func priority sd

If the demand is smaller than the oldest available stock of the single-deep lanes ($d < s_{qitsd}$) remove of as many stock as d_{it} ($s_{qitsd} = s_{qitsd} - d$ and $d = 0$).

If the demand is bigger than the oldest available stock of the single-deep lanes ($d > s_{qitsd}$) remove all the stock ($s_{qitsd} = 0$ and $d = d - s_{qitsd}$).

If there is only s_{qitmdk} available ($s_{qitsd} = 0$ and $s_{qitmdk} > 0$):

Func priority md

If the demand is smaller than the oldest available stock of the muli-deep lanes ($d_{it} < s_{qitmdk}$) remove of as many stock as d_{it} ($s_{qitmdk} = s_{qitmdk} - d$ and $d = 0$).

If the demand is bigger than the oldest available stock of the multi-deep lanes ($d_{it} > s_{qitmdk}$) remove all the stock ($s_{qitmdk} = 0$ and $d = d - s_{qitmdk}$).

Update the inventory variables: s_{sd}; l_{sd}; s_{md}; l_{md}; s_{qitsd}; l_{qitsd}; s_{qitmdk}; l_{qitmdk}; b_{qitmdk}; c_{qitmdk}; k_{qitmdk}.

Step 2. Repeat *Step 1* until all the demand is satisfied and supplied ($d = 0$).

Calculation of the indicator: This is the key element of the model to find out the best possible solution in terms of space losses. Six alternative formulas have been stablished to calculate the indicator, all related with the space losses (Honeycomb losses and accessibility losses) and space utilisation.

Indicator	Formula	Units
I1	Honeycomb losses (Horizontal) + Aisle losses	[m^2]
I2	Honeycomb losses (Horizontal + Vertical) + Aisle losses	[m^2]
I3	Honeycomb losses (Horizontal) + Aisle losses + Structure space	[m^2]
I4	Honeycomb losses (Horizontal + Vertical) + Aisle losses + Structure space	[m^2]
I5	Aisle losses + Structure space	[m^2]
I6	Item volume / (Structure volume + Aisle volume losses)	[%]

Table 2. Indicators of the Model

Below are the formulas to calculate the indicators:

- ***Honeycomb losses (horizontal):***

$$\left(k - c_{q_{it}itmdk}\right) \cdot (r_l \cdot r_d) \tag{6}$$

- ***Honeycomb losses (vertical):***

$$\left(\frac{b_{q_{it}itmdk} - c_{q_{it}itmdk} \cdot z}{z}\right) \cdot (r_l \cdot r_d) \tag{7}$$

- ***Aisle losses***

$$(l_{itsd} + l_{itmd}) \cdot \left(\frac{a}{2}\right) \cdot (r_l) \tag{8}$$

- ***Structure space***

$$\left(l_{itsd} + \sum_{k=1}^{K} l_{itmdk} \cdot k\right) \cdot (r_l \cdot r_d) \tag{9}$$

- ***Item volume***

$$s_{it} \cdot (r_l \cdot r_d \cdot r_h) \tag{10}$$

- ***Structure volume***

$$\left(l_{itsd} + \sum_{k=1}^{K} l_{itmdk} \cdot k\right) \cdot z \cdot (r_l \cdot r_d \cdot r_h) \tag{11}$$

- ***Aisle volume losses***

$$(l_{itsd} + l_{itmd}) \cdot \left(\frac{a}{2}\right) \cdot (r_l \cdot r_h) \tag{12}$$

4.2. RELOCATION

The objective of "The Relocation" process is to relocate the articles to a specific depth k even though it is not the optimal solution individually. First of all, it is required to count the number of each lane depth during all the horizon time taken into consideration during the analysis and select the final depth

k candidates of the warehouse (*k-list*). Then, the process performs the same process as "Local Optimization" but instead of trying all possible k ($k = 2,..,max\_k\_depth$), it iterates through the specific k (*k-list*) most used in the solution of "Local Optimization".

Pre-process: In addition to calculating the variable of the maximum number of lanes of single-deep, the process calculates the *k-list* to make sure the algorithm explores the most suitable depths based on the results of the Local Optimization process that principally depend on the characteristics of the incoming and the demand of all the products.

Step 1. Count the number of each lane depth during all the horizon time.

$$k_{candidate} = \sum_{i=1}^{I}\sum_{t=1}^{T} k_{q_{it}mdk} \quad \forall \mathrm{k} \tag{28}$$

Step 2. Taking into consideration the number of areas (N) with multi-deep lanes ($k > 1$), create the *k-list* with the N maximum $k_{candidate}$.

4.3. REDISTRIBUTION

The **Redistribution process** follows three main steps to relocate the products, the "Initial Peak Redistribution" process, the "Backward Redistribution" process and the "Forward Redistribution" process. In this specific process, the approach is substantially different from the one utilised during the Local Optimization process and the Relocation process. Even though the Incoming Distribution process is the basis of the performance of the three main steps, the indicator is not the decisive element of the decision. In the redistribution process, the key element in choosing between the different alternatives is the space utilisation of the warehouse. The main objective in this process is to reduce the space utilisation due to the redistribution of the incoming products. Follow next steps:

Step 1. (Initialisation) Read the data of the results of the "Relocation" process. Set the *k-list* and the inventory results of all the products: s_{sd}; l_{sd} ; s_{md}; l_{md} ; s_{qitsd} ; l_{qitsd}; s_{qitmdk}; l_{qitmdk}; b_{qitmdk}; c_{qitmdk}; k_{qitmdk}.

Step 2. Carry out the "**Initial Peak Redistribution"** process:

Step 2.1. Determine the day with the maximum number of multi-deep lanes of depth k ($k > 1$). Set the variables of the *depth* and *time*: k and t.

Set the variable *product*: $i = 1$.

Step 2.2.

If product i at day t has incoming (q_{it}) and if incoming (q_{it}) is located in a k depth lane execute the **"Redistribution Iteration"** process.

Step 2.2.1. Relocate incoming (q_{it}) to single-deep rack following the **case 1** of the **Incoming Distribution** process.

Calculate the new Space utilization (Space_utilisation_2.2.1 = Space utilisation)

Step 2.2.2. Relocate incoming (q_{it}) to single-deep and multi-deep lanes of depth k different than the k studied following the **case 2** of the **Incoming Distribution** process.

Calculate the new Space utilization (Space_utilisation_2.2.2 = Space utilisation).

Repeat Step 2.2.3. until there are no other possibilities to explore (All k options in *k-list*).

Step 2.2.3. Relocate incoming (q_{it}) to multi-deep lanes of depth k different than the studied k following the **case 3** of the **Incoming Distribution** process.

Calculate the new space utilization (Space_utilisation_2.2.3 = Space utilisation).

Repeat Step 2.2.3. until there are no other possibilities to explore (All k options in *k-list*).

Step 2.2.4. Relocate incoming (q_{it}) to multi-deep lanes of depth k different than the studied k and single-deep lanes following the **case 4** of the **Incoming Distribution** process.

Calculate the new space utilization ((Space_utilisation_2.2.4 = Space utilisation).

Repeat Step 2.2.4. until there are no other possibilities to explore (All k options in k-list).

Step 2.2.5. Relocate incoming (q_{it}) to multi-deep lanes of depth k different than the studied k following the **case 5** of the **Incoming Distribution** process.

Calculate the new space utilization.

Repeat Step 2.2.5. until there are no other possibilities to explore (All k options in k-list).

If the new space utilisation is smaller than the current space utilisation: Select the new space utilisation as the current space utilisation and update the k-list and the inventory variables (Output = New space utilisation).

Update the product: $i = i + 1$

Repeat *Step 2.2.* for all the products, until there are no more possibilities to explore ($i = I$).

Repeat *Step 2.1.* until there are no more depth k to explore in the *k-list*.

Step 3. Repeat *Step 2* while the Space utilisation calculated at the end of Step 2 is smaller than at the beginning (*Output* < *Input*).

Step 4. It starts the **"Backward redistribution"** process: Determine the maximum number of multi-deep lanes of depth k of the k-list. It starts with the first depth k of multi-deep lanes and saves the depth k and day t with this maximum number of lanes.

Set the variable of the *time* and the *backward position* (var): *var; t = day t* with the maximum number of multi-deep lanes of depth $k - var$.

Step 4.1. Execute *Step 2.1.* for all the products.

Update the variable of the backward *position*: $var = var - 1$.

Step 4.2. Set the variable of the *time*: $t = day\ t$ with the maximum number of multi-deep lanes of depth k.

Execute *Step 2*.

Step 5. Repeat *Step 4* until the variable of the *time* in *Step 4.1.* is smaller than 0 ($t < 0$) in all depths k.

Step 6. It starts the **"Forward redistribution"** process: Determine the maximum number of multi-deep lanes of depth k of the k-list. It starts with the first depth k of multi-deep racks and saves the depth k and day t with this maximum number of lanes. Set the variable of the *time* and the *forward position* (var): *var; t =*

day t with the maximum number of multi-deep lanes of depth k + var;

Step 6.1. Execute *Step 2.1.* for all the products. Update the variable of the *forward position*: *var = var + 1*.

Step 6.2. Set the variable of the time t = *day t* with the maximum number of multi-deep lanes of depth k.

Execute *Step 2* for all the products.

Step 7. Repeat *Step 6* until the variable t in *Step 6.1.* is bigger than T (t > T) in all depths k.

5. VALIDATION OF THE MODEL

To test the model, four experiments are carried out, with different types of database that variates the amount and the frequency of incoming with a similar demand. It is thus easier to test four types of warehouse with the following features:

- Experiment 1: A low volume and a low rotation of stock.
- Experiment 2: A low volume and a high rotation of stock
- Experiment 3: A high volume and a low rotation of stock.
- Experiment 4: A high volume and a high rotation of stock.

Results I1	Only single deep	Improvement I1
195,15 m^2	230,18 m^2	15,22 %
335 p	300 p	-10,45 %
45,69 %	51,03 %	-10,46 %
0,78 p/m^2	0,67 p/m^2	14,10 %
161.226,88 €	161.887,5 €	0,41%
481,27 €/p	539,63 €/p	10,81%

Table 3. Comparison of the best results of the Model with Only single-deep lanes of the Experiment 1.

Results I5	Only single deep	Improvement I5
454,04 m^2	498,71 m^2	8,96 %
795 p	650 p	22,31 %
41,07 %	47,31 %	-13,19 %
0,72 p/m^2	0,62 p/m^2	16,13 %
320.069,37 €	298.256,25 €	7,31 %
402,60 €/p	458,86 €/p	12,26 %

Table 4*. Comparison of the best results of the Model with Only single-deep lanes of the Experiment 2.*

Results I1	Only single deep	Improvement I1
688,79 m^2	1.127,86 m^2	38,93%
1.630 p	1.470 p	10,88%
57,89%	62,96 %	-8,05%
1,37 p/m^2	0,82 p/m^2	67,07%
500.296,25 €	617.748,75 €	19,01%
306,93 €/p	420,24 €/p	26,96%

Table 5*. Comparison of the best results of the Model with Only single-deep lanes of the Experiment 3.*

Results I4	Only single deep	Improvement I4
1.370,90 m^2	1.898,94 m^2	27,81 %
2.690 p	2.475 p	8,69 %
69,01 %	69,90 %	-1,27%
1,35 p/m^2	0,91 p/m^2	48,35 %
905.751,25 €	1.000.321,88 €	9,45 %
336,71 €/p	407,81 €/p	17,43 %

Table 6*. Comparison of the best results of the Model with Only single-deep lanes of the Experiment 4.*

6. CONCLUSION

The Optimization Model for multi-deep storage systems is evaluated through four different scenarios with different kinds of stock and rotation. The following insights can be put forward:

1. Primarily, the final results of the four simulations carried out are better in terms of space utilisation than a storage system consisting of only single-deep lanes. With these results, can be confirmed that the model gives a properly result. The improvements in terms of space utilisation are at least 8,96% (*Experiment 2: A low volume of stock and a high rotation*) compared with single-deep storing system. Total space utilisation reductions are even better with other simulations, where the Model achieves reductions of 38,93% (*Experiment 3: A high volume of stock and a low rotation*). Based on the results, it can be deducted that the Model performs better with low rotation of stock than for high rotation of stock and for high volumes of stock than for ow volumes of stock. However, it does not mean that the Model will always give better results as more simulations are required to confirm this result.
2. After the realisation of the experiments, there have not been results indicating a progress in the implementation of the “Forward Redistribution Process”. The design of this process was based on very specific circumstances where there were two peaks of usage of lanes of the same depth k and the second peak was a stumbling block for the improvement of the redistribution process.
3. In all the experiments the indicator 6 has outperformed the results of the other indicators. This indicator is not suitable for the model created in this Master Thesis. It is important to highlight that it does not mean that the cube utilisation is a bad formula when the lane depth of a storage system is decided upon, it means that it does not fit with the process followed by this model but could work properly through the application of another approach.
4. Further research development is expected on the other multi-deep storing system like shuttle racking that allow to store different SKU i to the same multi-deep lane. This way the storage efficiency of the storing system could increase significantly compared with the results obtained with the drive-in storing systems.

REFERENCES

[1] Accorsi, R., Baruffaldi, G. and Manzini, R., 2017, *“Design and manage deep lane storage system layout. An iterative decision-support model”,* Int J Adv. Manuf. Technol. 92:57–67.

[2] Bartholdi, J. J., III and Hackman, S. T., 2017, *“Warehousing & Distribution Science”*, Release 0.98, www.warehouse-science.com.

[3] Cormier, Gilles, and Eldon A. Gunn, 1992, *"A review of warehouse models"*, European Journal of Operational Research 58(1), 3-13.

[4] Goetschalckx, M., and Ratliff, H.D., 1991, *“Optimal lane depths for single and multiple products in block stacking storage systems,"* IIE Transactions, 23(3), 245-258.

[5] Kay GM, 2009, *“Lecture Notes for Production System Design”*. Department of Industrial and Systems Engineering, North Carolina State University.

[6] Kind DA, 1965, *“Measuring warehouse space utilization”,* Transportation and Distribution Management 7(5):23–33.

Table of Contents

1. Introduction

1.1. Background

The problem considered in this paper is taken from a real-life issue. Conventional models in designing storage systems assume uniform and deterministic incoming and demand in specific quantities and time intervals. The adopted storage infrastructures significantly influence the warehouse performance in terms of space and time efficiency in addition to the initial investments costs that can differ substantially.

Logflow is a consulting company that offers master warehouse operations solutions in the entire logistics, the hard automation and IT automation. In order to improve their current services in the design of suitable storage systems adapted to each client and its needs, they are looking for scientific research to build an optimization model for multi-deep storage systems..

1.2. Problem statement

For storage of pallets, single-deep pallet racking is the most flexible solution, each pallet being individually reachable. However, it is not always the optimal solution when taking into consideration other factors as space utilization. One of the possible solutions to increase storage density are multi-deep storage systems (floor storage, drive-in racking, shuttle racking etc.).

The objective of this Master Thesis is to build a calculation and optimization model for multi-deep storage systems. The purpose is to compare multi-deep with single-deep pallet storage systems and to find out the optimal depth or combinations of depths in order to get the maximum space efficiency.

1.3. Overview

The model will take into consideration the product and environment parameters as the stock size, the reception size, the stock rotation in the warehouse, stackability (possible number of levels in the building), etc. The output of the model should predict the fill rate relative to the gross number of pallet positions and to the floor surface, as well as the resulting cost per net pallet location. The model has to aid of the design of new storage systems from green field determining the optimal combination of single-deep and multi-deep of different depth of a warehouse when flow of products is stochastic and dynamic in nature.

2. Warehouse

This chapter is an overview of the warehouse concept, how they add value to the operations of a business, the challenges that a warehouse has to face in its design phase and the main elements that have to be taken into consideration.

2.1. Introduction

Warehouses are a key aspect of modern supply chains and play a vital role in the success, or failure, of businesses today (Frazelle, 2002a). Warehouses perform a pivotal function and are an essential component of any supply chain (Gu et. al 2007) to establish smooth and efficient logistic operations. A warehouse is the point in the supply chain where raw materials, work-in-process (WIP), or finished goods are stored for various lengths of time (Kay, 2015).

The current competitive global trend requires more product variety and short response times, which means new requirements for warehousing operations (Accorsi, et. al 2017). In recent years, enterprises have completely reconfigured their supply chain to address increasing customer service levels and demand variability. Their major roles include:

- **Balance supply with customer demand**: Warehouses form the key nodes in supply chain networks decoupling demand from supply in time and quantity (Zaerpour et. al 2011). Variations in demand try out the capacity of a supply chain because demand can change quickly but supply chain needs more time to change. On one hand, the role of a warehouse is managing the increase in demand thanks to having stock product. On the other hand, they can work as a buffer slowing or holding inventory back from the market in case the decrease in demand. Thus, warehouses help to respond quickly and effectively when demand changes and/or batching production and transportation.
- **Consolidation of products:** In order to provide combined delivery to customers and leverage transportation fixed costs. Thanks to warehouse is easier to fill the capacity and subsequently the distributors can offer to vendors large transportation freight for downstream customers. On the other hand, it is also useful for the distribution companies that can receive downstream easily and manage drivers shifts easily. In addition, it allows to value added-processing such as kitting, pricing, labelling and product customization (Gu et. Al 2007). To sum up, a warehouse allows products to be collected, sorted, and distributed efficiently (Kay, 2015).

In this section it has been explained the main functionalities of a warehouse. Considering its critical impact on customer service levels and logistics costs, as well as the degree of complexity involved, it is thus imperative to the success of businesses that warehouses are designed so that they function cost effectively (Baker et. al 2009).

2.2. Warehouse design

This section is a brief review of the main warehouse design challenges, the majority of the logistic costs of a warehouse are determined during the design phase (B. Rouwenhorst et al., 1999). The five major decisions (Gu et al., 2010) are illustrated in *Figure 1*: Determining the overall structure, sizing and dimensioning of the warehouse and its departments; determining the detailed layout within each department; selecting warehouse equipment; and selecting operational strategies.

Figure 1*. Five major decisions during the design phase of a warehouse.*

- **The overall Structure (or conceptual design):** Determines the functional departments, the quantity of storage departments, the technologies that will be used and the definition of the main process of the supply chain. The principal issues to be undertaken are the minimization of costs (taking into consideration possible investments and operating costs) and to comply with all the storage and throughput requirements.
- **Sizing and dimensioning of the warehouse:**
  - **Warehouse sizing:** Determines the storage capacity of a warehouse based on the inventory levels and costs (warehouse construction, storage and products within the warehouse and the demand not satisfied due to the storage capacity).
  - **Warehouse dimensioning:** It is the translation of storing capacity into floor space with the purpose to evaluate and estimate the construction and operating costs.
- **Determining the detailed layout with each department:** The main storage problems are organized as:

- Pallet block-stacking pattern, i.e. storage lane depth, number of lanes for each depth, stack height, pallet placement angle with regards to the aisle, storage clearance between pallets, and length and width of aisles.
- Storage department layout, i.e. door location, aisle orientation, length and width of aisles, and number of aisles.
- AS/RS configuration, i.e. dimension of storage racks, number of cranes.

- **Selecting warehouse equipment:** The grade of automation in a warehouse, the type of storage and the handling systems that will be employed. This strategic choice will affect broadly the other decisions such as the overall warehouse investment and performance
- **Selecting operational strategies:** Long term decisions that stay on the operation strategies that will affect the overall system. The most important operation strategies are storage policies and related to the order picking.

This section has explained the main warehouse design challenging problems. This Master Thesis focuses on the second of the categories introduced above because it is about the selection of the sizing and dimensioning of the storage system that implies high investments for the companies.

2.3. Storage policies

Storage is a major warehouse function that allows product to be available where and when it's needed. The storage policy is the set of rules which determines where each item of an incoming product must be stored. Hereunder, there are the three main storage policies (Kay, 2015) for multiple SKU's employed to select storage locations:

- **Dedicated (or Fixed):** Each SKU has a predetermined number of pallet locations assigned to it and only that product may be stored there; the capacity assigned to each SKU has to be at least equal to the maximum stock. The main advantages of this policy are the minimizing of handling costs and the simple tracking of the SKU's. However, the building costs increase considerably.
- **Shared (or Open or Floating or Random):** All SKUs can be stored in the same lanes; the idea is to assign a product to more than one storage location. In that way when one location becomes empty, it is available for reassignment perhaps to a different product. Thus, each SKU can be stored in any (usually the closest) available slot. In this case, the total capacity of all the locations must be at least the storage space corresponding to the maximum on-hand stock of all the SKUs. The building costs are minimised thanks to the improving the space efficiency but the handling costs increase drastically due to high time-consuming of activities and the tracking of the SKU's become much more difficult than dedicated storage because the location of each SKU has to be saved by a warehouse management system for retrieval activities.

- **Class-based:** A combination of dedicated and random storage policies, where each SKU is assigned to one of several different storage classes. Classes normally are classified by the on-hand stock correlation between different SKUs. On this occasion, the building and handling costs are between dedicated and random policies.

2.4. Storage systems

2.4.1. Single-deep storage systems

The main benefits of moving a pallet from floor storage into pallet rack are the reduction of labour costs by improving the efficiency of storing and retrieving and the creation of additional pallet positions. Thanks to the rack supports each pallet is independently accessible and can be retrieved at any level of the rack. Moreover, it also helps to the protection of the product form damaged and improves the security of the warehouse by avoiding the unstable pallet stacks. Although the use of single-deep pallet racking eliminates the honeycomb losses, additional space is required for vertical and horizontal rack members.

2.4.2. Multi-deep storage systems

With the fast development of e-commerce business and competitiveness, the requirement for compact storage area with high throughput capacity and flexible system structure, has increased a lot. However, before the implementation of a new handling system that needs a major investment, many factors should be investigated by analytical or simulation models.

If we want to increase the capacity of a storage, you can store the pallets back-to-back away from the pick aisle, this concept is known as the multi-deep storage systems. Using this system, we achieve more space efficiency because pallet positions can share the same aisle space, the aisle space doesn't provide storage, it only provides accessibility to the forklift to insert or extract a pallet. Therefore, aisle space is reduced to the minimum to guarantee these operations because it is not revenue-generating space.

Figure 2. *Multi deep storage system.*

Multi-deep storage systems are normally used to store homogeneous products, with large numbers of pallets with the same reference. Generally, the infrastructure is made up of storing racks, that create interior paths to shift loads, with support rails for the pallets. The forklifts get in these paths with the load above the level it will be located.

It's important to bear in mind that the entire footprint of a lane is normally reserved for a SKU if this lane is already storing a pallet because if there was more than one SKU stored in the same lane, some pallets could be double-handled during the retrieval operation, which could outweigh any space saving achieved by the multi-deep storage system.

There are several possibilities to implement a multi-deep storage system but the most commonly used are:

- **Floor storage** is the simplest way of storing pallets arranged in lanes over the floor with only one level of height.
- **Drive-in racking** enables the forklift to drive inside the rack frame to access the interior pallets and perform the put-away and retrieval functions from inside the rack frame.
- **Shuttle racking** are composed of shuttle carriers that get inside the rack frame and perform the put-away and retrieval functions from the same aisle without losing the wasting time that the forklift need to get inside the rack frame.

This research is going to focus on **drive-in** racking systems.

2.5. Storage locations and handling equipment

This section is a presentation about the most important parameters of the storage locations and the handling equipment that are taken into consideration in this Master Thesis.

2.5.1. Pallet racking

In this research we are going to centre our efforts in the products that flows in the supply chain as pallets. The pallet is the largest standardized material-handling unit, which is a rigid base on which unit loads can be allocated (almost all of them are made of wood, but some are made of durable plastic). In logistics, it is commonly used the term SKU (Stock keeping Unit) as the smallest physical unit of a product that is traced by an organization, thus, a pallet is going to be considered a SKU.

The universal system to store pallets are the pallet racks, used normally for bulk storage accommodating reasonable sized slots. The main advantage of pallet racking is the accessibility and the stackability (number of levels of the rack), on one side it offers greater accessibility because the pallets can be reached independently and on the other side it offers higher stackability because the height of the rack can be configured and adjusted depending on the loads and the warehouse ceiling limitations.

The standardized dimensions of the pallets appear in *Table 1*.

Stringer length x deckboard length	**Most prevalent in**
1219 x 1016 mm	North America
1000 x 1200 mm	Europe Asia
1165 x 1165 mm	Australia
1067 x 1067 mm	North America, Europe, Asia
1100 x 1200 mm	Asia
800 x 1200 mm	Europe

Table 1. *The six standard sizes of pallet, from ISO Standard 6780: Flat pallets for intercontinental material handling – Principal dimensions and tolerances. (The stringers are the supports underneath that are spanned by the deckboards).*

Being meticulous, it is important to take into consideration the minimum clearances required by the different racking systems. Basically, they depend on the load, the maximum height and obviously the rack model. Single-deep and multi-deep rack minimum clearances are different.

i. Single deep

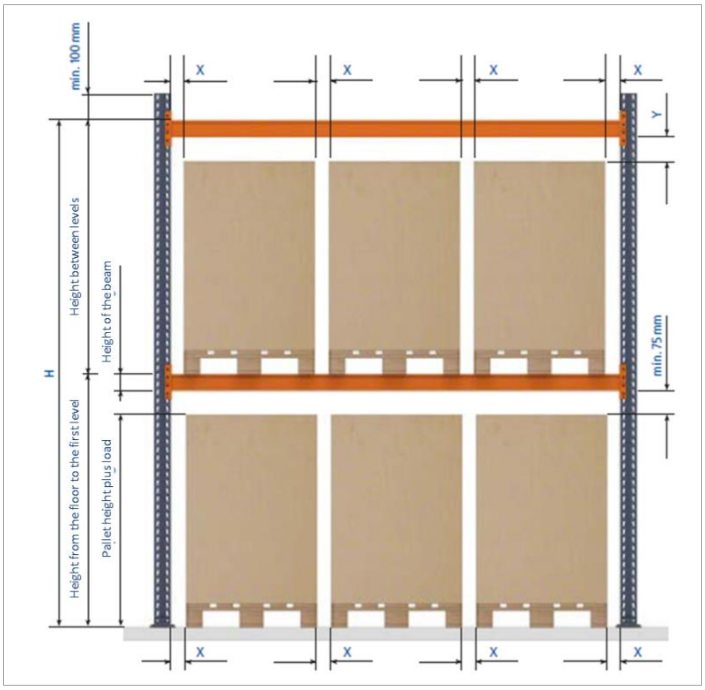


Figure 3. *Minimum X and Y clearances for single-deep rack.*

Tolerances and clearances between the gap:

Y: Height between the pallet and the bottom of the beam for levels different to the altitude +0.

X: Minimum clearance between the pallets or loads.

Range between levels (mm):	**Class 400**		**Class 300A**		**Class 300B**	
	X	Y	X	Y	X	Y
0 ≤ H ≤ 3.000	75	75	75	75	75	75
3.000 ≤ H ≤ 6.000	75	100	75	75	75	100
6.000 ≤ H ≤ 9.000	75	125	75	75	75	125
9.000 ≤ H ≤ 12.000	100	150	75	75	100	150
12.000 ≤ H ≤ 13.000	100	150	75	75	100	175
13.000 ≤ H ≤ 15.000	-	-	75	75	100	175

Table 2. Minimum X and Y clearances between pallets.

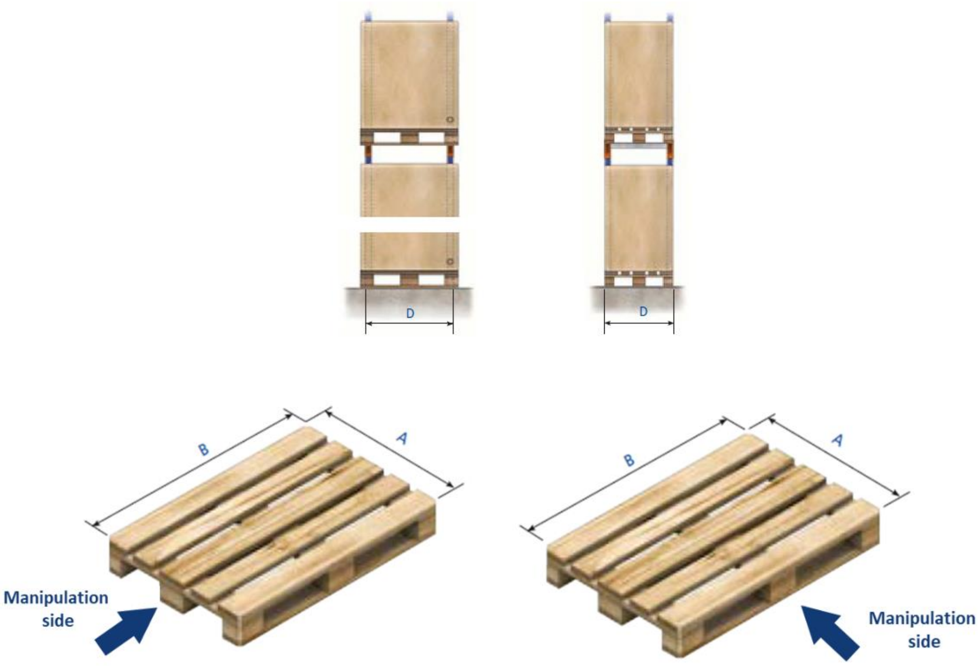


Figure 4. Narrow side vs Wide side manipulation

Measurment of the length of the frame (mm)		
Pallets manipulatd by the narrow side	Measurments of the pallets	Pallets manipulated by the wide side
D = 1.100	800 x 1.200	D = 800
D = 1.100	1.000 x 1.200	D = 1000
D = 1.100	1.200 x 1.200	D = 1.200

Table 3. Measurements of the length of the pallet depending on the manipulation side.

ii. Multi-deep

Height *(Figure 5)*

These are the minimum clearances to consider:

- F: Bottom and intermediate levels height = pallet height plus load + 150mm.
- G: Top level height = pallet height + 200mm.
- H: Total height = the sum of all the levels.

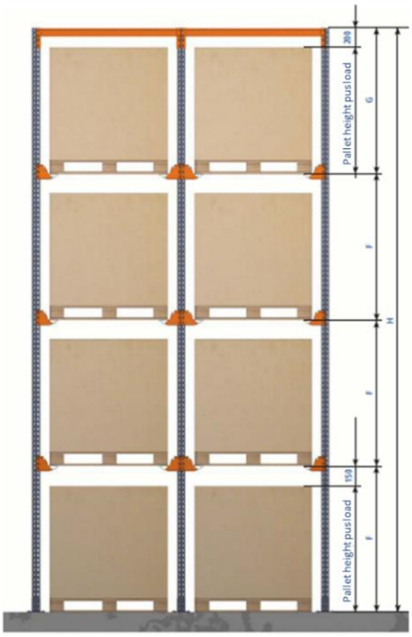


Figure 5. *Minimum height clearances for multi-deep rack.*

Depth *(Figure 6)*

The minimal measurements of depth to consider are the following:

- X: Sum of the depth of all the pallets (considering the load if it outlays) plus a clearance for unit load, of minimum, 25mm.

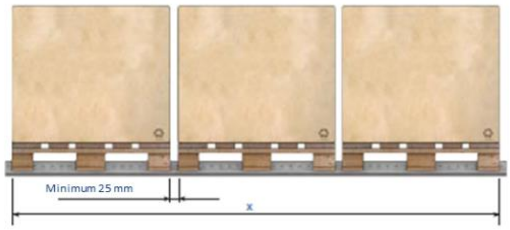


Figure 6. *Minimum depth clearance for multi-deep rack.*

2.5.2. Handling equipment

The most common tool used to get access to SKUs in the pallet racks are the lift trucks. It is true that due to the evolution of technology there have emerged new automated storage and retrieval systems (AS/RS), however, the cost and investment that companies have to face to deploy these systems may not offset the economic effort. The most common lift trucks are:

- Counterbalance lift truck.
- Reach and double-reach lift truck.
- Turret truck.
- Combitruck.

Single-deep		
Description	**Reachtruck**	**Combitruck**
Application	For pallet storage and internal transport	Narrow aisle warehouse
Minimum aisle width	3.000 mm	1.900 mm
Minimum headers width	3.000 mm	5.500 mm
Target price	45.000 €	90.000 € + 1.500€ per asile
Multi-deep		
Description	**Reachtruck (drive-in)**	**Reachtruck (drive-in)**
Application	For pallet storage and internal transport	Narrow aisle warehouse
Minimum aisle width	3.000 mm	3.000 mm
Minimum headers width	3.000 mm	-
Target price	45.000 €	45.000 € + 25.000€ per shuttle

Table 4. *Handling equipment data provided by Logflow.*

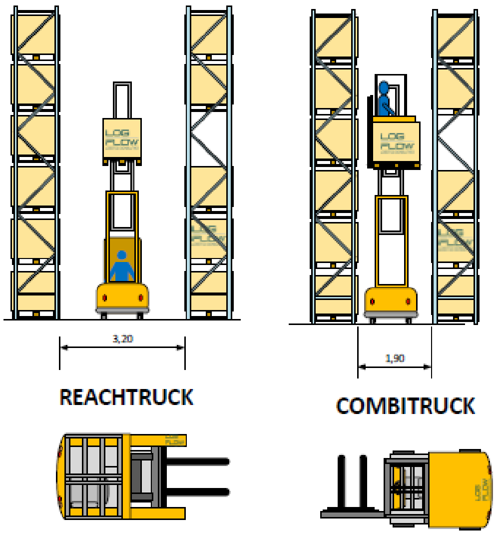


Figure 7. *Aisle required and performance of reachtruck and combitruck lift trucks.*

The importance of this section remains in the aisle width needed for the lift truck because it will constrain the honeycombing effect and subsequently the results of the optimization model.

2.6. Lane depth

When the pallets are stored in a multi-deep storage system they are arranged in lanes, the depth of a lane is the number of pallets stored back-to-back away from the aisle. As it has previously been mentioned, when the storage policy is *dedicated storage* the whole footprint of a lane is reserved for a specific pallet if it is already storing a pallet.

Multi-deep storage systems allow to increase the effective utilization of space. However, deeper lane depths produce more pallet locations but they are of less value because some of the pallets are not directly accessible. The first depth lane will only be accessible when the interior pallets locations become available. To illustrate this dichotomy let's see the example in and *Table 5* :

	Single-deep storage	Double-deep storage	Difference
Storage locations	120	160	33,33 %
Directly accessible locations	120	80	-33,33 %

Table 5 *.Example of space used by single deep and double deep storage systems*

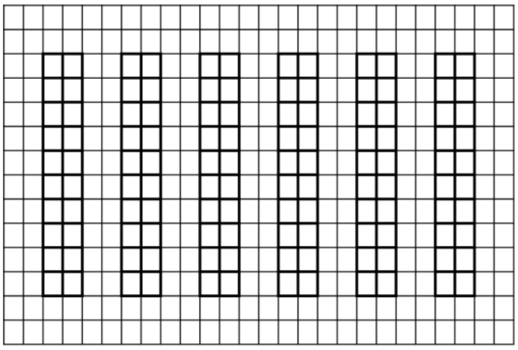


Figure 8*. Single deep storage layout*

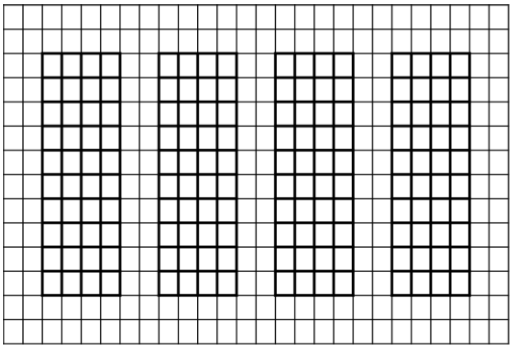


Figure 9. *Double deep storage layout*

While in the single-deep rack storage systems all the pallets are directly accessible, which allows to reassign as soon as the current pallet is shipped, in the multi-deep storage systems only the first lane depth will be accessible. To explain mathematically these losses, we adopt the warehouse science explanation. The space efficiency is measured by two metrics named accessibility and honey combing (Bartholdi et. al 2017).

2.6.1. Honeycomb losses

As it has been explained, multi-deep storage systems can prompt some items not being accessible. This kind of losses are commonly named honeycomb losses, the price paid for accessibility, is the unusable empty storage space in a lane or stack due to the storage of only a single SKU in each lane or stack since storing items from different SKUs would block access. Otherwise, if a SKU is stored in a single-deep storage system, honeycomb losses disappear since the depth and height of the slot can exactly match the storage space need for the SKU and the SKU is available at any moment.

The empty space that can be observed in the following picture are the “horizontal” and “vertical” honeycomb losses

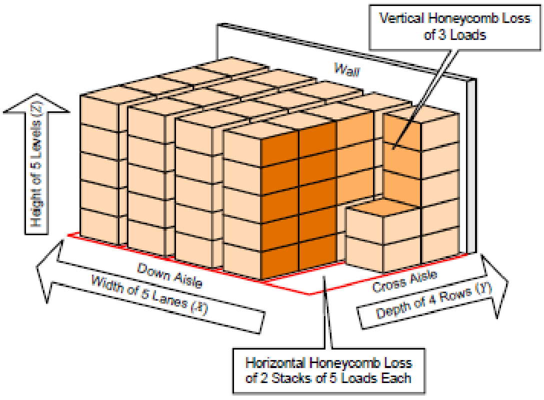


Figure 10. *Horizontal and Vertical Honeycomb effect*

2.6.2. Accessibility losses

The accessibility space is the section of the aisle needed to access the pallet with the handling equipment and it is taken into consideration as the half of the width multiplied by the length of one rack location

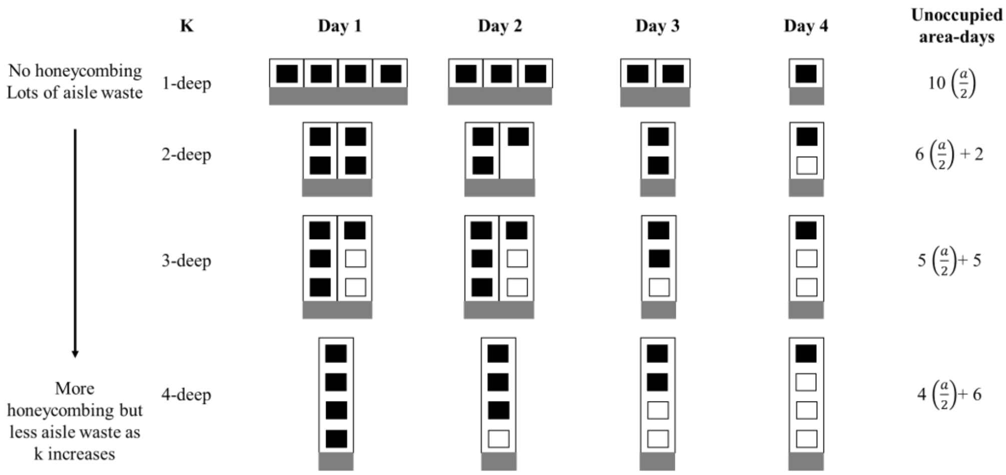


Figure 11. *Honeycomb and aisle losses effect*

To elaborate on the honeycomb losses and the accessibility losses, look at the *Figure 11. Honeycomb and aisle losses effect*, and imagine that the demand is one pallet per day, so that one pallet location is extracted constantly during four days. If there are four pallets at Day 1 they can be stored in four different combinations. If *a* is the width of the aisle, measured as a fraction of the length of a pallet position, in single-deep racks the aisle losses at the end of the period will be 10·(a/2), much bigger than if they are located in a multi-deep rack of 4-depth lanes, 4·(a/2). However, the honeycomb effect in the single-deep racks is null, neither vertical or horizontal honeycomb effect appear because at every day t all the pallets are accessible and as more depth rows appear in a multi-deep storage system the honeycomb effect increases gradually.

To maximize space efficiency, following the logical and information provided by accessible and honeycomb losses the SKU *i* should be stored in the k-depth that minimizes these losses, so the k-depth that allows to minimize the *surface x time* consumption. Evaluating all possibilities with a known aisle width and rack dimensions it is easy to determine which would be the best k-depth lane for this SKU *i*.

Nevertheless, this case was so easy to estimate because it was only one height level (the vertical honeycomb effect is null), the demand was constant and there was only one SKU *i* in the warehouse. But when these variables change, the incoming and demand is not constant, the levels of the rack are bigger and there are multiple SKUs, there are several combinations and it becomes extremely complicated to choose which is the best depth lane for the warehouse. This Master Thesis is going through a model that helps to solve this challenge.

In this chapter, there has been summarised the main function and features of a warehouse, focusing on the storing systems that are taken into consideration in this model. To go on with the definition of the model, in next chapter there is a literature review of the existing investigation and to understand the most important elements that will have to be implemented in the model.

3. Literature review

In this chapter we are going to see the proposed models and research carried out in the past related to the optimization of multi-deep storage systems. The literature has lengthy debated the issues of warehouse design and management, which is aimed at minimizing the operation costs and time and increasing the supply chain performance.

3.1. Lane depth

The determination of the optimal lane depth in multi-deep storage system has a great impact on layout problems and has been widely studied. There are a number of methods for increasing the utilization of storage space. For the space efficiency objective of this Master Thesis, the determination of the optimal lane depth for the products is the central issue, as a storage lane remains unavailable for arriving pallets until its current content has been totally depleted by demand, thereby creating the need to optimize storage lane depth (Cormier et. al 1992). With this goal, most of the following literature review relates to the maximization of the utilization of storage space and specifically to the optimal lane depth.

Starting from the beginning of the research related to the lane depth. In 1965 it was the first time that the optimal lane depth for single product was determined considering the floor space utilization (Kind 1965, 1975). Then, it was proposed a new model to balance the required number of lanes including the put-away and retrieving activities (Kooy 1975). This model introduced the idea that an optimal warehouse system should have an empty lane available of the depth required to store an arriving product optimally. A GPSS simulation model was created to determine if some performance indicators as primary storage area and aisle area are affected by the effect of alternate lane depths on volume utilization (Marsh 1979).

Some years later, there was an improvement determining a single depth for all the stored products (Matson et. al 1981, 1984). This model was mostly focused in the space utilization objective. Ten years later, a better procedure was developed determining the optimal number of lanes and optimal lane depths for single products following a triangular pattern and comparing the results with several heuristics (Goetschalckx et. al 1991).

The concept of cube-utilization index is defined as the space needed to store the products including the accessibility space (Kay et. al 2009). In this research they explored how the cube utilization aids the determination of the optimal lane depth in presence of dedicated and random storage configurations They determined that the lane depth that maximizes the cube utilization corresponds to the best compromise between honeycomb loss and down-aisle space. Cube utilization is the percentage of the total space required for storage actually occupied by the loads being stored.

$$Cube\ utilization = CU = \frac{item\ sapce}{total\ space}$$

$$CU = \frac{item\ space}{item\ space + (honeycomb\ loss) + (down\ aisle\ space)} \tag{1}$$

$$CU\ (3-D) = \frac{x \cdot y \cdot z \cdot \sum_{i=1}^{N} M_i}{TS\ (D)} \tag{2}$$

$$CU\ (2-D) = \frac{x \cdot y \cdot \sum_{i=1}^{N} \left\lceil \frac{M_i}{H} \right\rceil_i}{TS\ (D)} \tag{3}$$

X = lane/unit-load with

Y = unit-load depth

Z = unit-load height

Mi = maximum number of units of SKU i

M = Maximum number of units of all SKUs

N = number of different SKUs

D = number of rows

TS(D) = total 3-D space (given D rows of storage)

TA(D) =total 2-D area (given D rows of storage)

To summarise the main patterns for the lane-depth mentioned before look at *Table 6. Extant lane depth models*. (Accorsi et al. 2016).

Notations	Reference	Model
	Kind 1975)	$k_i = \sqrt{\frac{q_i a}{l z_i} \cdot \frac{a}{2l}}$
qi: incoming batch sizo of SKU i	Mason and White (1981)	$k_i = \sqrt{\frac{(q_i + 2Ii)a}{2l z_i}}$
q incoming batch sizo for generic SKUs		
zi: stackability of SKU i	Goetschalckx and Ratliff (1991)	$k_j = \frac{ja}{2l}$
ki: lane depth of SKU i		
Ii:lane depth for generic SKUs		
l: pallet length of SKU i	Kay (2009)	$k = \left\lfloor \sqrt{\frac{a([2max(qi)]-n)}{2nlz}} + \frac{1}{2} \right\rfloor$
a: aisle width		
I: 1,..,n SKUs	De Koster (2010)	$k = \sqrt{\frac{a}{ln} \sum_{i=1}^{n} q_i}$
j: 1,..,m lanes		
	Bartholdi and Hackman (2013)	$k_i = \sqrt{\frac{a}{2} \cdot \frac{q_i}{z_i}}$

Table 6. *Extant lane depth models.*

Last decade, the research focused on defining the lane depth for multiple products linked with the relocation problem, (Kim et. al 2006). They focused on determining the depth of a storage row to store a particular lot of product under random storage conditions, for multiple products stored in lanes of the same depth, as well as locating a single lot of product to existing lane depths. The objective always is to minimize the average amount of storage space required. A more complex cost model was developed incorporating cost of travel and the cost of storage space (White et al. 2013) and studied the impact of changes in the unit load and the turnover rate.

All of these patterns take into consideration a constant incoming and demand with the same number of SKU per lot in all the horizon time. This can be applicable in some cases, but in the majority of cases, it is far away from the real-world warehousing systems. To face a seasonable demand, a decision-support model (ILP) was investigated, to the optimal lane depth, storage mode and zone or the assignment of the incoming products (Accorsi et al. 2016). However, there has not been research for random incoming and demand. In this Master Thesis, this issue will be dealt with, to find out the best possible combination of single-deep and multi-deep lanes for random demand and production.

4. Optimization of a multi-deep storage system model

This chapter is going to present the model for the optimization of a multi-deep storage system. First of all, section 4.1. explains the terms and assumptions that cements the framework of the model and section 4.2. shows the performance and the process and procedures that the model follows to get the best combination of single deep and multi-deep storage systems.

4.1. Problem statement

This Master Thesis studies the multi-deep against the single-deep storage systems. Hereafter there are listed the main assumptions and specifications that are taken into consideration.

First, the scenario of this Master Thesis is to design a new warehouse from green field according to a specific inventory mix and the expected incoming and production of the different products, it is indispensable to know at least two of this three variables to go on with the model execution. Even though, the setting data can be provided by the client in units of product, in kilos, in boxes, in packages… it has always to be switched to pallets before starting to run the model.

Second, batch loads enter and leave the system on a First In, First Out (FIFO) basis by shipment. However, when the pallets are stored in a multi-deep rack, they follow a Last In, First Out (LIFO) basis. Thus, applying this methodology, the first product that gets in the warehouse is the first one that gets out and, moreover, it deals with the design of multi-deep racks that are accessed in a LIFO manner as “drive-in” rack. The restriction of drive-in racks is that pallets are not independently accessible when the handling operations of replenishing/emptying are required.

Third, different products and different batches of the same product are not mixed in the same lane of a multi-deep rack although different batches of the same product can be mixed in the same lane of a single deep rack. This procedure is followed to ensure that there will not be quality deterioration caused by double handling. However, the same product can be distributed in different lanes so long as it doesn’t mix with other products or different shipments of the same product.

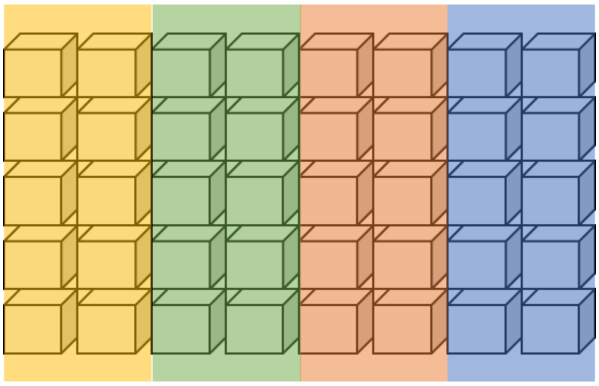


Figure 12. One reference to each lane.

The inventory control system is based on the following assumptions:

- Both for incoming quantities (from production or suppliers) and outgoing quantities (to satisfy customer demands). There is not a fixed quantity per cycle time, it changes from cycle to cycle and is a random data that does not follow any pattern.
- These are the main practical considerations that add a degree of complexity that the previous research had not yet considered.
- Both for incoming and outgoing orders, the entire order is received in one batch and the replenishment is instantaneous.
- The stock of day *t* is the is the sum of the stock of day *t-1* and the incoming of day *t* minus the demand of day *t*.

Related to the variability of lanes of different depth available in the warehouse, it is not necessary that all the lanes have the same depth. There can be a few of them based on the requirement of the client. It will be possible to choose the number of areas with different k-depth lanes. This restriction most of the times is predefined if there is already the structure of the building. In any case, it is possible to select the number of areas available in the warehouse.

To define the height of the storage systems there are two possibilities, it can be determined by the height of the warehouse assuming that you can use all the space in this area to locate the storage system or it can be chosen by the client. In both cases always following the regulations and normative specified in the section *2.5.1 Pallet racking.*

4.2. Notations

Below, there are listed the variables that are used in the model. (It is important to remark that the name of the variables cannot match with the variables used in the executable *.py* program).

Symbol	Description	Units
i: 1,..,I	SKUs.	u
k	Lane depth.	pallet
t: 1,..,T	Time *t* with a given horizon *T*.	day
q_{it}	Expected incoming of SKU *i* at time *t* (batch).	pallet
d_{it}	Expected demand of SKU *i* at time *t*.	pallet
s_{it}	Stock of SKU *i* at time *t*.	pallet
$d_{q_{it}t}$	Demand of the incoming of SKU *i* at time *t* (q_{it}).	pallet
$s_{q_{it}}$	Stock of the incoming of SKU *i* at time *t* (q_{it}).	pallet
$s_{q_{it}\,sd}$	Stock of the incoming of SKU *i* at time *t* (q_{it}) stored in a single deep lane.	pallet
$s_{q_{it}\,mdk}$	Stock of the incoming of SKU *i* at time *t* (q_{it}) stored in a multi-deep lane of *k-depth*.	pallet
$b_{q_{it}\,mdk}$	Stock of the incoming of SKU *i* at time *t* (q_{it}) stored in a multi-deep lane of k-depth that full partially a lane.	pallet
$c_{q_{it}\,mdk}$	Number of rows of the incoming of SKU *i* at time *t* (q_{it}) stored in a multi-deep lane of k-depth that full partially a lane.	row
$k_{q_{it}\,mdk}$	Depth of the lane of the incoming of SKU *i* at time *t* (q_{it}) stored in a multi-deep lane of *k-depth*.	row
$l_{q_{it}\,sd}$	Lanes of the incoming of SKU *i* at time *t* (q_{it}) stored in a single-deep lane.	u
$l_{q_{it}\,mdk}$	Lanes of the incoming of SKU *i* at time *t* (q_{it}) stored in a multi-deep lane of *k-depth*.	u
l_{itsd}	Lanes of SKU i at time *t* of single-deep lanes.	u
l_{itmdk}	Lanes of SKU *i* at time *t* of multi-deep lanes of k-depth.	u
a	Accessibility space required by each lane.	m
z	Stackability levels of the storage system.	pallet
N	Number of different areas in the warehouse.	u
r_l	Length of the rack.	m

r_d	Depth of the rack.	m
r_h	Height of the rack.	m
c_{sd}	Cost per pallet position of a single-deep rack.	€ /pallet position
c_{md}	Cost per pallet position of a multi-deep rack.	€ / pallet position
c_s	Cost per square meter.	€ / m^2
c_h	Cost of the handling equipment.	€

Table 7. *Data and variables of the optimization of a multi-deep storage system*

It is important to remark the difference between d_{it} and $d_{q_{it}t}$, while d_{it} is the demand that will have product i at day t, so the number of pallets that will be retrieved of the warehouse to satisfy the demand, by contrast, $d_{q_{it}t}$ is the number of pallets of the batch q_{it} that will be retrieved at day t.

4.3. Constraints

This section shows the main boundaries and constrains of the model:

- Guarantee the number of lanes occupied by the incoming SKU i at day t is not higher than the empty lanes in day t.
- Ensure that the multi-deep storage system has a limited number of areas with lanes of different depth.
- Guarantee that a SKU i cannot be mixed with another SKU i at the same lane.
- Ensure that a q_{it} (incoming of SKU i at day t) cannot be mixed with another q_{it} of the same SKU i at another day t.
- The model is only prepared to determine the best combination of single-deep and multi-deep lanes for drive-in storage systems (Floor storage can be determined with this model setting a z=1); Shuttle racking cannot be determined due to the third and fourth constraint).
- The model is not prepared to conduct stock breakdown situations, it is necessary to prepare the incoming and demand data at the beginning to ensure that this will not happen.

4.4. Multi-deep storage system model

In this section, it is showed all the process step by step that the model uses to calculate the best combination of single-deep and multi-deep lanes of a storage system. We propose a recursive model that assigns the incoming products to the optimal lane depth by minimizing the space losses generated by accessibility and honey combing inefficiencies. Multiple parameters contribute to the lane assignment, not only the rack sizes and the stackability levels, but also the quantity and the frequency of the incoming batches and the demand, that varies day by day.

The suggested model follows a bottom-up strategy that is composed of three main process. The first process consists of solving the problem in a local scope, handling each product *i* separately and independently. The best possible number of single-deep or multi-deep lanes for each product can thus be found. The output from this local analysis is usually a solution with a large number of areas with lanes of different depths. Then, to find a more holistic and better global solution, the model proceeds to a relocation of the products in a particular number of areas with a specific depth based on the results of the previous process. Hence, to find a global solution, it transfers the output from a large number of areas with lanes of different depths to a solution with a particular number of areas with a specific depth. Finally, the model focusses on determining which will be the best solution globally.

The methodology used to approach the final solution is divided in the following three main processes: The **"Local Optimization",** the **"Relocation"** and the **"Redistribution"** process.

Figure 13. *Multi-deep storage system model process.*

The first process called **"Local Optimization"** approaches a solution for each specific product that allows to give an output of the combination of single-deep and multi-deep lanes for each day *t* and each product *i*. The second process called **"Relocation"** converts the current output into a combination of a specific number of different multi-deep lanes with different depths with a more limited scope. It changes from a wide range of multi-deep lanes into a few multi-deep lanes in function of the requirements of the client. Finally, the third process called "**Redistribution"** is focused on balancing the final solution relocating the products into other single-deep or multi-deep lanes thanks to taking into consideration the other lanes needed by the other products. The aim of this process is the optimization of the total space utilisation of the storing system and its output is the final solution of the model which gives a definitive combination of single-deep and multi-deep lanes for all days *t* and all the products *i*.

The input data of the model are:

- Height of the warehouse (pallet locations).
- Aisle width (m).
- Rack width, length and height per location of multi-deep storage system (m).
- Demand per product (pallets).
- Production per product (pallets).
- Horizon time (t = 1,..,T).
- Number of areas available (N).
- Number of Products (i = 1,..,I).
- Costs (€):
    - Cost per pallet location single-deep.
    - Cost per pallet location multi-deep.
    - Cost per square meter of the floor of the warehouse (including building, utilities, etc.).
    - Cost of handling equipment.

The output data of the model are:

- The **combination of single-deep and multi-deep lanes of different depth**: The most suitable storage system in terms of space efficiency, it is defined by the number of single-deep lanes and the number of multi-deep lanes of depth *k*.

$$l_{sd} = max \sum_{i=1}^{I} l_{itsd} \quad \forall\, t \tag{4}$$

$$l_{mdk} = max \sum_{i=1}^{I} l_{itmdk} \quad \forall\, t \tag{5}$$

- The **surface of the warehouse (m^2):** It comes from adding the space required by the storage system structure to the aisle space required for the lift trucks to manoeuvre inside the warehouse.

$$surface\ of\ the\ warehouse = structure\ surface\ + aisle\ surface$$

$$structure\ surface = \left(l_{sd} + \sum_{k=1}^{K} l_{mdk} \cdot k\right) \cdot (r_l \cdot\ r_d) \tag{6}$$

$$aisle\ surface = \ \left(l_{sd} + \sum_{k=1}^{K} l_{mdk}\right) \cdot r_l \cdot\ a \tag{7}$$

- The **storage capacity (pallet locations)**: The total amount of pallet locations that the storage system provides.

$$Storage\ capacity = \ l_{sd} \cdot\ z + \left(\sum_{k=1}^{K} l_{mdk} \cdot k\ \right) \cdot z \tag{8}$$

- The **storage efficiency (%)**: The fill rate relative to the gross number of pallet locations. It is calculated with the average of pallet position occupied (stock at day *t*) divided by the pallet position available (Storage capacity).

$$storage\ efficiency_t = \frac{\sum_{i=1}^{I} s_{it}}{Storage\ capacity}\ \forall\ t \tag{9}$$

$$storage\ efficiency = \frac{storage\ efficiency_t}{T} \cdot 100 \tag{10}$$

- The **surface efficiency (pallet/m²):** The fill rate relative to the floor surface.

$$surface\ efficiency_t = \frac{\sum_{i=1}^{I} s_{it}}{surface\ of\ the\ warehouse}\ \forall\ t \tag{11}$$

$$surface\ efficiency = \frac{surface\ efficiency_t}{T} \tag{12}$$

- The **cost of the warehouse (€):** It is calculated based on the cost per pallet location of the different storage systems (multi-deep or single-deep), the cost of the floor (m^2) that will be required to build the structure of the storage system and the cost of the handling equipment (lift truck).

$$cost\ of\ the\ warehouse = surface\ of\ the\ warehouse\ cost + storage\ system\ cost + handling\ ewuipment\ cost$$

$$surface\ of\ the\ warehouse\ cost = surface\ of\ the\ warehouse \cdot c_s \tag{13}$$

$$storage\ system\ cost = l_{sd} \cdot z \cdot c_{sd} + \left(\sum_{k=1}^{K} l_{mdk} \cdot k\right) \cdot z \cdot c_{md} \tag{14}$$

$$handling\ equipment\ cost = c_h \tag{15}$$

- The **cost per net pallet location (€/pallet)**: It is calculated dividing the cost of the warehouse by the storage capability.

$$cost\ per\ net\ pallet\ location = \frac{Cost\ of\ the\ warehouse}{storage\ capacity} \tag{16}$$

4.4.1. Local Optimization

i. Description

"Local Optimization" is the first process applied in the model to get a local solution for each product in which it will be possible to know the best possible combination of single-deep and multi-deep lanes of different depth for each product *i* at each day *t* to minimize the space losses.

The model will work in a recursive way (see *Figure 15*) exploring all the possible combinations of lane depth that could be used in a hypothetic case and compares the value of an indicator to find out which is the best option.

ii. Input / Output

To determine the scope of the "Local Optimization" process it is required the following data of the *Figure 14*. It is very important to remind that the output of this process is focused on find the best possible local solution, not the best global possible solution.

***Figure 14**. Input and Output of the Local Optimization process.*

iii. Procedure

To carry out this process the algorithm will perform the next steps:

Step 0. (Initialization) Set the number of products that have to be stored in the warehouse ($i=1,..I$), the rack dimensions (r_l, r_h *y* r_d), the stackability (z), the aisle width (a) and the costs variables (c_{sd}, c_{md}, c_s and c_h).

Step 1. Read the initial data of product i to determine the q_{it} (incoming of product i at day t), the q_{it} (demand of product i at day t), the horizon time ($t = 1,..,T$) and $d_{q_{it}}$ (consumption of the batch q_{it} at day t).

Step 2. Carry out the **Pre-Process** to calculate the maximum number of single-deep lanes (*maximum lanes sd*) and maximum depth (*maximum k-depth*).

Step 3. Initialization of the variables of the *time*, the *stock* and the *Indicator_def*: $t = 1$; $s_{sd} = 0$; $l_{sd} = 0$; $s_{md} = 0$; $l_{md} = 0$; $s_{q_{it}sd} = 0$; $l_{q_{it}sd} = 0$; $s_{q_{it}mdk} = 0$; $l_{q_{it}mdk} = 0$; $b_{q_{it}mdk} = 0$; $c_{q_{it}mdk} = 0$; $k_{q_{it}mdk} = 0$; $Indicator\_def = M$.

Step 4. If at day t the incoming of product I is bigger than 0 ($q_{it} > 0$) execute the **"Incoming Distribution"** process.

Step 5. If at day t the demand of product i is bigger than 0 ($d_{it} > 0$) execute the **"Inventory Control"** process.

Step 6. The algorithm moves to next day. Update the *time, the stock* and the *Indicator_def*: $t = t + 1$; s_{sd}; l_{sd}; s_{md}; l_{md}; $s_{q_{it}sd}$; $l_{q_{it}sd}$; $s_{q_{it}mdk}$; $l_{q_{it}mdk}$; $b_{q_{it}mdk}$; $c_{q_{it}mdk}$; $k_{q_{it}mdk}$; $Indicator\_def = M$.

Step 7. Repeat *Step 1* to *Step 6* for all the products. Update i: $i = i + 1$.

Step 8. The process finishes when there are no more products to store in the warehouse (i=I).

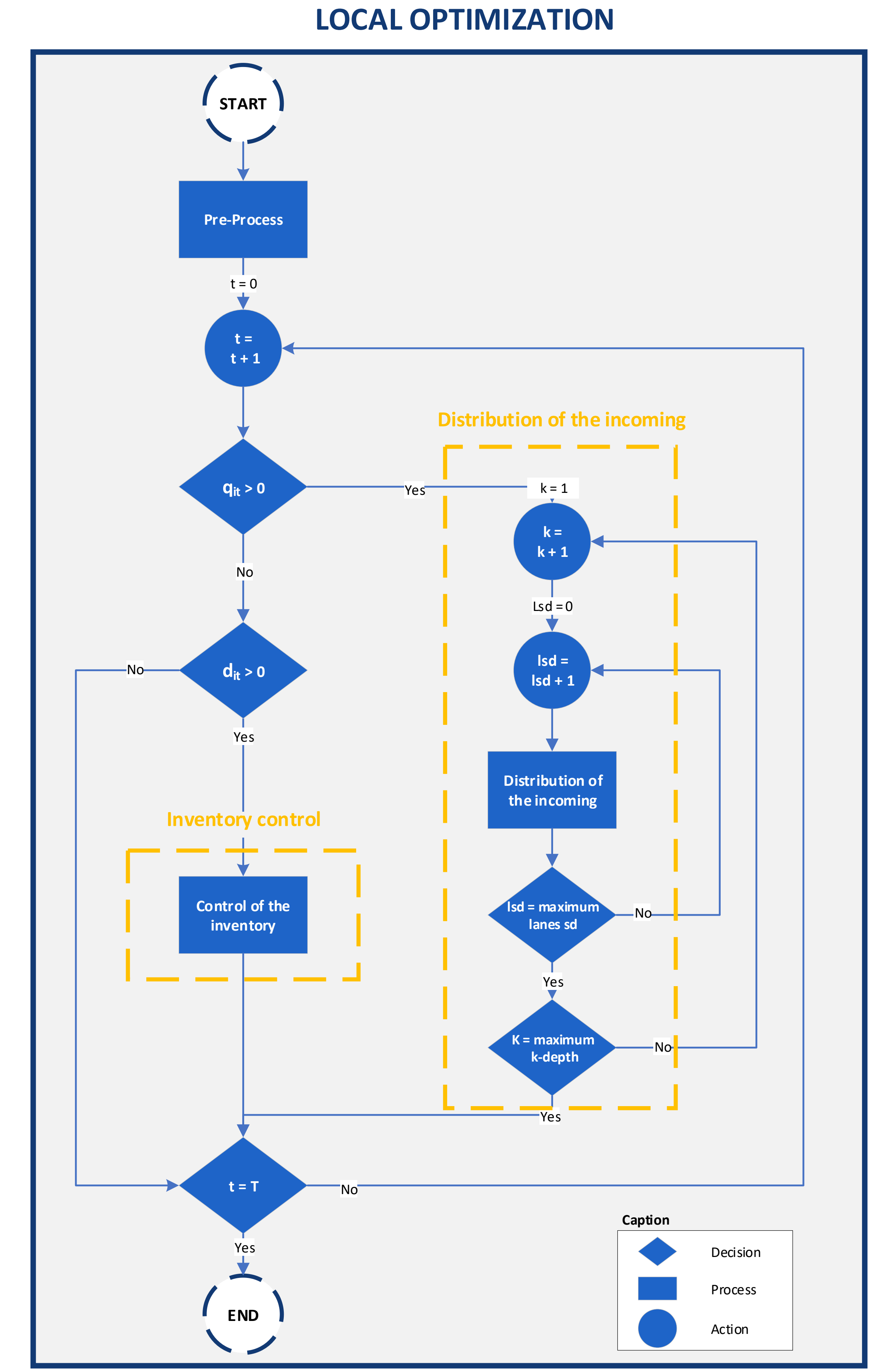


Figure 15. Local Optimization Process

Looking at the Local Optimization more thoroughly, the three main activities carried out by the algorithm are the "**Pre-process**", the **"Incoming Distribution"** and the **"Inventory Control"**. These are outlined below:

1. **Pre-process**: To be able to analyse all possible alternatives, it is indispensable to calculate two variables called the *maximum number of single-deep lanes* and the *maximum k-depth.* In addition to make sure the algorithm covers all the candidate solutions, it will set the limits of the exploration.
    - The **maximum number of single-deep lanes** (*maximum lanes sd*) that you would need if you only had single-deep storage racks available in your warehouse. With this variable it is guaranteed that the model will always give the possibility to locate any quantity of incoming of the product in a single-deep rack. The calculation is the following:

$$s_{it} = \begin{cases} q_{it} - d_{it} & if\ \ t = 1 \\ s_{it-1} + q_{it} - d_{it} & if\ t\ = 2, .., T \end{cases} \tag{17}$$

$$maximum\ stock\ level_i = max\, s_{it} \tag{18}$$

$$maximum\ lanes\ sd_i = \frac{maximum\ stock\ level_i}{z} \tag{19}$$

    - The **maximum lane depth** (*maximum k-depth*) that you would need if you only had multi-deep storage racks available in your warehouse. With this variable it is ensured that the model will be able to locate any quantity of incoming of the product in all the possible depth *k* combinations. The computation is the following:

$$maximum\ incoming_i = max\, q_{it} \tag{20}$$

$$maximum\ k - depth_i = \frac{maximum\ incoming_i}{z} \tag{21}$$

    With the maximum number of single-deep lanes and the maximum lane depth it is ensured that the model will try all the possible combinations of single-deep and multi-deep racks.

2. **Incoming distribution:** This process is only applied in case the incoming of product *i* at day *t* is bigger than 0 *(q_{it} > 0)*. It is understood as incoming any product *i* that has to be stored in the warehouse as well as it comes from internal production as well it comes from suppliers. The issue is not where it comes from, it is that there is a product that has to be stored in the warehouse.

   The main objective of this process is to decide where to locate the incoming product *i*, to a single-deep lane, to a multi-deep lane or to both of them in the most efficient way in terms of floor space utilisation. To ensure the model chooses the best option it explores the five different potential options and calculates the indicator for all of the options: For all the different depths (*k = 2,..,maximum k-depth*) and for all the possible number of single-deep lanes (*lsd = 1,..,Maximum lanes sd*).

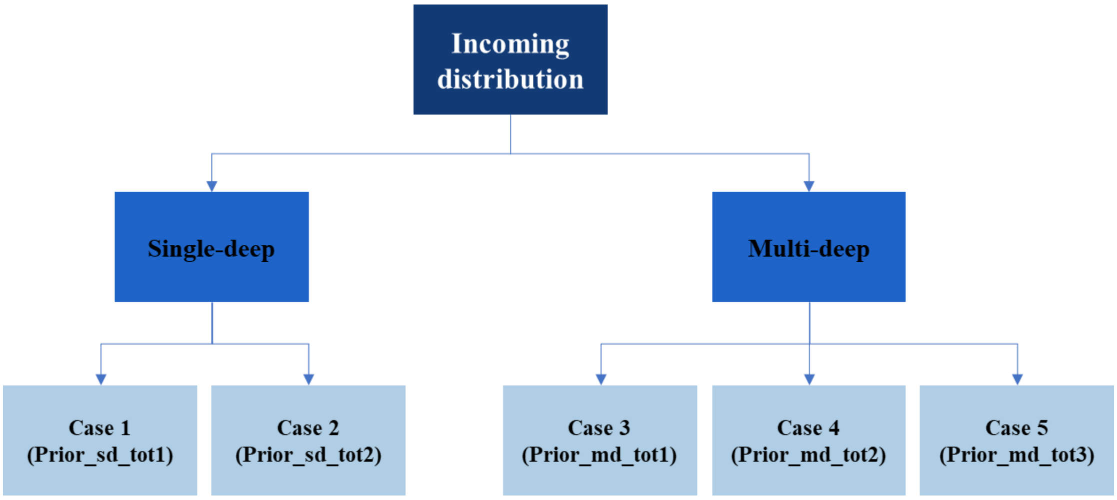


Figure 16. *Incoming distribution possibilities.*

Step 0. (Check q_{it}): If at day *t* the incoming of product *i* is bigger than 0 ($q_{it} > 0$) execute the **"Incoming Distribution"** process.

Step 1. It starts the iterative process:

Set the variable *k*: $k = 2$

Step 1.1. Update *k*: $k = k + 1$

Set the variable *lsd*: $lsd = 0$

Step 1.2. Update *lsd*: $lsd = lsd + 1$

Determine the possible combinations of single-deep and multi-deep lanes and the stock distribution in the different lanes following the variables *k* and *lsd*.

Case 1 ("Prior_sd_tot1"): All the q_{it} is stored in single-deep racks.

Calculate the Indicator (*Indicator_Prior_sd_tot1 = Indicator*).

Case 2 (***"Prior_sd_tot2")***: All the q_{it} is stored in single-deep racks. But, if there is not enough space in the single-deep racks, the remaining q_{it} is located in multi- deep racks.

Calculate the Indicator: (*Indicator_Prior_sd_tot2 = Indicator*).

Case 3 ("Prior_md_tot1"): All the q_{it} is stored in multi-deep racks.

Calculate the Indicator: (*Indicator_Prior_md_tot1 = Indicator*).

Case 4 ("Prior_md_tot2"): All the q_{it} is stored in multi-deep racks. But, if there is some pallet of the q_{it} that full partially a multi-deep lane and it is smaller than the available space of the single-deep racks, then it is relocated to a single-deep rack.

Calculate the Indicator: (*Indicator_Prior_md_tot2 = Indicator*).

Case 5 (“Prior_md_tot3”): All the q_{it} is stored in multi-deep lanes. But, if there is some pallet of the q_{it} that full partially a multi-deep, then it is relocated to a single-deep rack (even though there could not be space, it would be necessary to add a new single-deep rack in the warehouse).

Calculate the Indicator (*Indicator_Prior_md_tot3 = Indicator*).

Select the minimum Indicator.

(Check the Indicator) If the Indicator is smaller than the *Indicator_def* (*Indicator < Indicator_def*) set the Indicator as the new *Indicator_def* and save the inventory variables (s_{sd}; l_{sd}; s_{md}; l_{md}; $s_{q_{it}\,sd}$; $l_{q_{it}\,sd}$; $s_{q_{it}mdk}$; $l_{q_{it}mdk}$; $b_{q_{it}mdk}$; $c_{q_{it}mdk}$; $k_{q_{it}mdk.}$).

Step 2. Repeat *Step 1.2.* until there are no more possibilities to explore (*lsd = maximum lanes sd).*

Step 3. Repeat *Step 1.1.* until there are not more possibilities to explore *(k = maximum k-depth).*

Too see more detail of the Incoming Distribution process go to Appendix *A.1. Incoming Distribution* where there is the mathematical algorithm.

To illustrate what are the different possibilities that the model offers to locate the incoming through the calculation of the indicator look at the following examples:

Database 1	
Description	**Data**
Lane depth (k)	3
Max_lanes_sd	8
q_{it}	37

Table 8. *Database 1*

Database 2	
Description	**Data**
Lane depth (k)	3
Max_lanes_sd	3
q_{it}	43

Table 9. *Database 2*

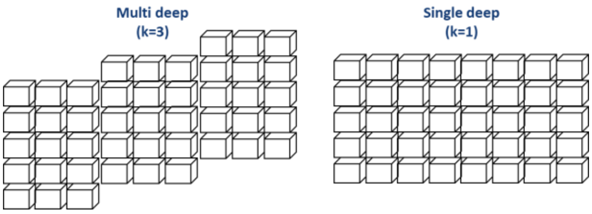
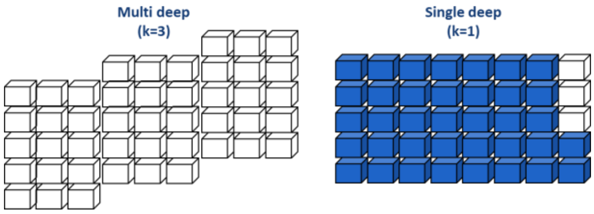
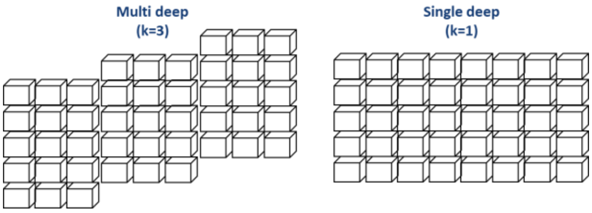
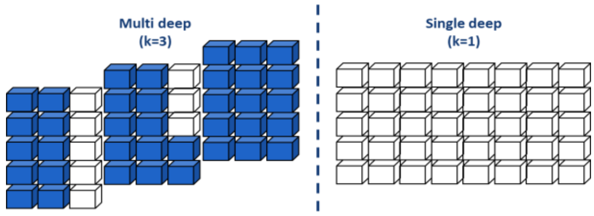
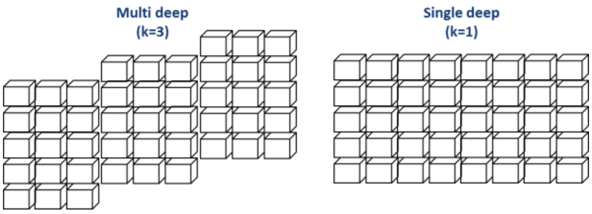
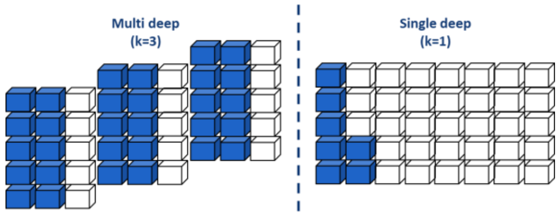
<table>
<tr><th colspan="2">1</th></tr>
<tr><td>Initial Inventory</td><td>Final Inventory</td></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">2</th></tr>
<tr><td>Initial Inventory</td><td>Final Inventory</td></tr>
<tr><td>N/A</td><td>N/A</td></tr>
<tr><th colspan="2">3</th></tr>
<tr><td>Initial Inventory</td><td>Final Inventory</td></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">4</th></tr>
<tr><td>Initial Inventory</td><td>Final Inventory</td></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">5</th></tr>
<tr><td>Initial Inventory</td><td>Final Inventory</td></tr>
<tr><td>N/A</td><td>N/A</td></tr>
</table>

Table 10. *Incoming distribution alternatives in function of the database 1.*

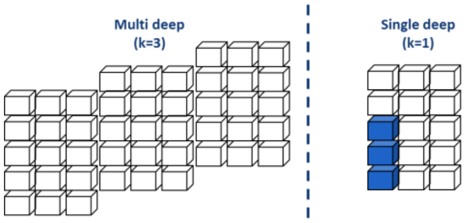
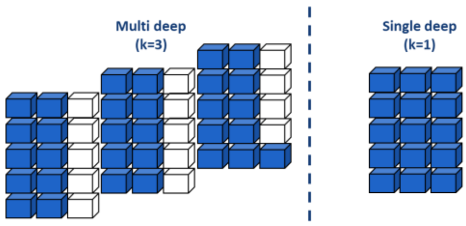
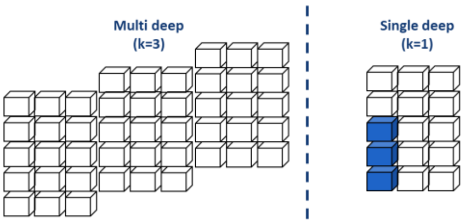
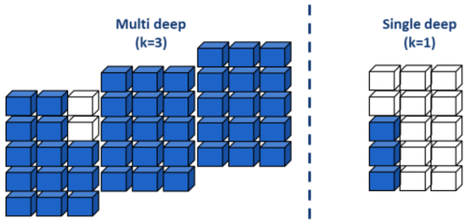
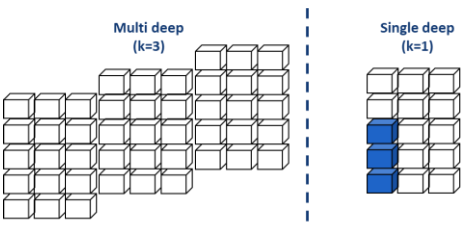
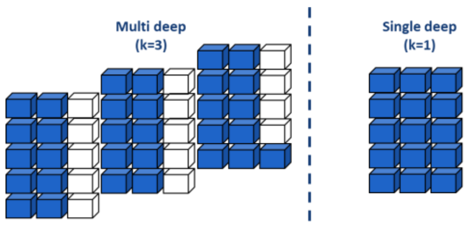
1	
Initial Inventory	**Final Inventory**
N/A	N/A
2	
Initial Inventory	**Final Inventory**
	
3	
Initial Inventory	**Final Inventory**
	
4	
Initial Inventory	**Final Inventory**
N/A	N/A
5	
Initial Inventory	**Final Inventory**
	

Table 11. *Incoming distribution alternatives in function of the database 2.*

With the methodology proposed in the Incoming distribution process, the algorithm explores all possibilities to store the incoming product *i*, not only storing in a single-deep or a multi-deep lane, but also in a shared way. Furthermore, it explores the five different possibilities explained above for all *depth* ($k = 1,..,K$) possibilities of multi-deep lanes and all possible number of lanes of single-deep lanes ($l_{sdi} = 0,..,maximum\ lanes\ sd$), this way, the model ensures that all possible combinations are taken into consideration.

3. **Inventory control:** This process is only applied in case the demand of product *i* at day *t* is bigger than 0. It is understood as demand as any consumption of product *i* that has to be removed from the warehouse.

   The main objective of this process is to decide which pallets of product *i* should be removed, from a single deep lane, from a multi-deep lane or from both of them. To ensure the model chooses the best option in terms of space efficiency the model explores the different potential options and calculates the indicator for all of the options. The main rule is to follow a FIFO (First In, First Out) policy to satisfy the demand. However, the possible emptying movements will increase depending on the distribution of the last q_{it} that still remains in the warehouse. There can be several options if the oldest q_{it} is stored single-deep and multi-deep lanes. Follow next steps to carry out the Inventory Control process:

Step 0. (Check d_{it}): If at day *t* the incoming of product *i* is bigger than 0 ($d_{it} > 0$) execute the **"Inventory Control"** process. Set $d = d_{it}$.

Step 1. (Check $s_{q_{it}\,sd}$ and $s_{q_{it}\,md}$):

Step 1.1. If there is $s_{q_{it}\,sd}$ and $s_{q_{it}\,mdk}$ available ($s_{q_{it}\,sd} > 0$ and $s_{q_{it}\,mdk} > 0$):

Func priority sd

If the demand is smaller than the oldest available stock of the single-deep lanes ($d < s_{qitsd}$) remove as many stock as d ($s_{q_{it}\,sd} = s_{q_{it}\,sd} – d$ and $d = 0$).

If the demand is bigger than the oldest available stock of the single-deep lanes ($d_{it} > s_{q_{it}sd}$) remove all the stock ($s_{q_{it}\,sd} = 0$ and $d = d - s_{q_{it}\,sd}$).

Calculate Indicator (*Indicator_priority_sd = Indicator).*

Func priority md

If the demand is smaller than the oldest available stock of the muli-deep lanes ($d < s_{q_{it}mdk}$) remove of as many stock as d ($s_{q_{it}mdk} = s_{q_{it}mdk} - d$ and $d = 0$).

If the demand is bigger than the oldest available stock of the multi-deep lanes ($d > s_{q_{it}mdk}$) remove all the stock ($s_{q_{it}mdk} = 0$ and $d = d - s_{q_{it}mdk}$).

Calculate Indicator (*Indicator_priority_md = Indicator)*

If Indicator priority sd is smaller than Indicator priority md execute ***Func priority sd***, otherwise execute the ***Func priority md***.

Step 1.2. If there is $s_{q_{it}\,sd}$ or $s_{q_{it}\,mdk}$ available ($s_{q_{it}\,sd} > 0$ or $s_{q_{it}\,mdk} > 0$):

If there is only $s_{q_{it}\,sd}$ available ($s_{q_{it}\,sd} > 0$ *and* $s_{q_{it}\,mdk} = 0$)*:*

Func priority sd

If the demand is smaller than the oldest available stock of the single-deep lanes ($d < s_{qitsd}$) remove of as many stock as d_{it} ($s_{q_{it}\,sd} = s_{q_{it}\,sd} – d$ and $d = 0$).

If the demand is bigger than the oldest available stock of the single-deep lanes ($d > s_{q_{it}sd}$) remove all the stock ($s_{q_{it}sd} = 0$ and d $d = d - s_{q_{it}sd}$).

If there is only $s_{q_{it}mdk}$ available ($s_{q_{it}sd} = 0$ *and* $s_{q_{it}mdk} > 0$):

Func priority md

If the demand is smaller than the oldest available stock of the muli-deep lanes ($d_{it} < s_{q_{it}mdk}$) remove of as many stock as d_{it} ($s_{q_{it}mdk} = s_{q_{it}mdk} - d$ and $d = 0$).

If the demand is bigger than the oldest available stock of the multi-deep lanes ($d_{it} > s_{q_{it}mdk}$) remove all the stock ($s_{q_{it}mdk} = 0$ and $d = d - s_{q_{it}mdk}$).

Update the inventory variables: s_{sd}; l_{sd}; s_{md}; l_{md}; $s_{q_{it}sd}$; $l_{q_{it}sd}$; $s_{q_{it}mdk}$; $l_{q_{it}mdk}$; $b_{q_{it}mdk}$; $c_{q_{it}mdk}$; $k_{q_{it}mdk}$.

Step 2. Repeat *Step 1* until all the demand is satisfied and supplied ($d = 0$).

To see more detail of the Inventory Control process, go to the *Appendix A.2 Inventory Control* where there is the mathematical algorithm.

INVENTORY CONTROL

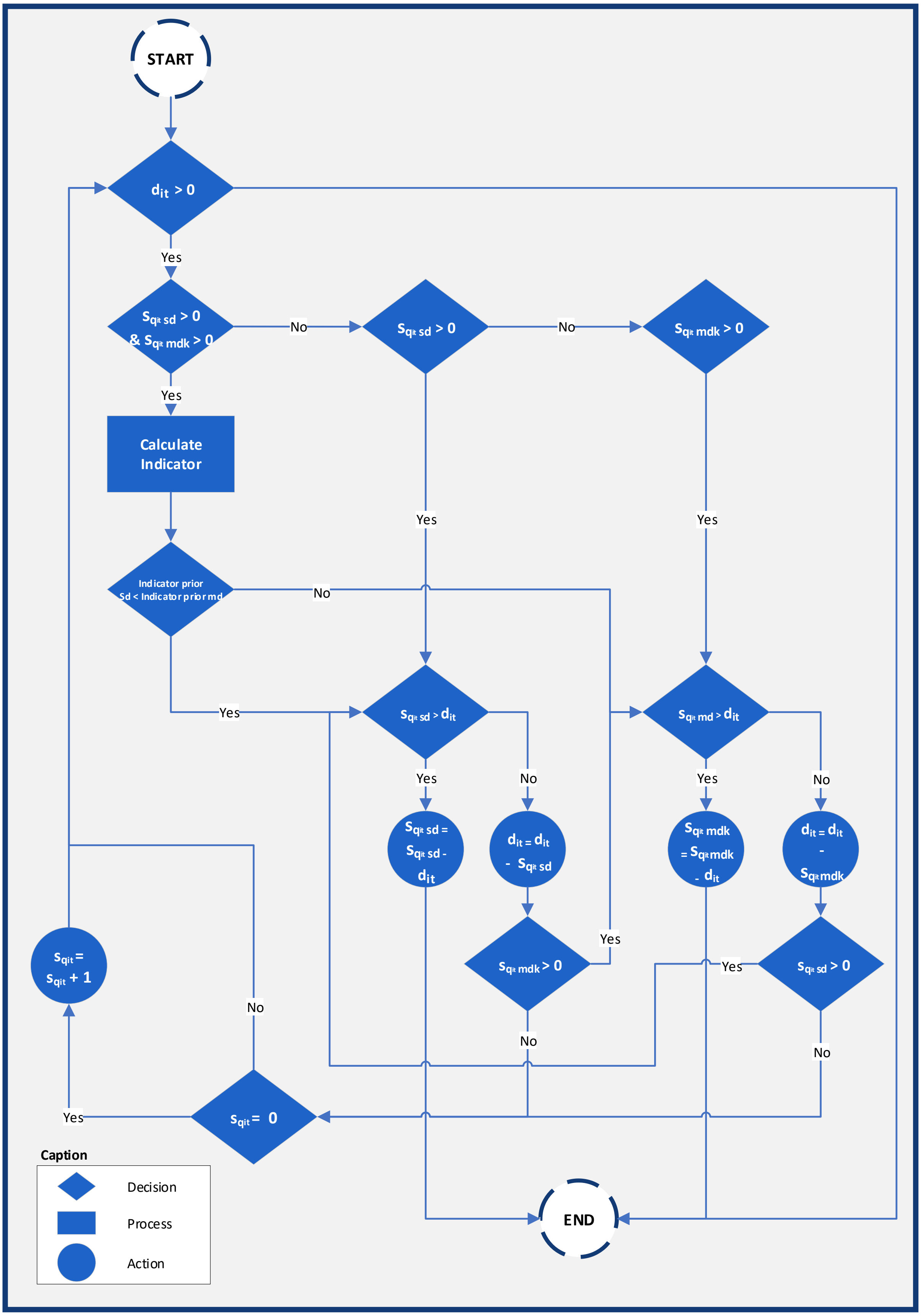


Figure 17. Inventory Control Process.

After the exposition of the most important process performed by the model it is explained how works, the calculation and the cements of the key element of the model, the indicator. It will determine which will be the best combination of single deep and multi-deep lanes of different depth. The indicator is the criteria selected to make the decision of storing and emptying a pallet location.

The indicator is based on the phenome of the "Honeycomb effect" and "Aisle losses" that produce space loses in a warehouse (for more detail see *2.6 Lane depth*). The "Honeycomb effect" can be summarised as the price paid for accessibility, is the unusable empty storage space in a lane due to the storage of only a single SKU *i* in each lane since storing items from another SKU *i+1* would block access to SKU *i*. The "aisle losses" is the section of the aisle needed to access the lane to carry out the replenishment and emptying movements.

As it is mentioned before, the complexity of this study is the application of the "Honeycomb effect" and "Aisle losses" in a real warehouse where the demand and the incoming don't follow any pattern or any cyclic distribution. However, the importance of using as indicator the space losses consist in the effect of **time** to the stock of the lot of product. The time effect can be traduced as the demand that following a LIFO inventory rotation will affect the inventory level along the time till the q_{it} is all consumed.

To introduce this effect of time in the calculation of the indicator it is necessary the amount of demand that will be consumed of each q_{it} each day *t*. Thanks to knowing two of the three variables demand, incoming or stock beforehand it is possible to calculate this consumption applying a FIFO policy. Then, it will be possible to calculate the indicator following the pròposed methodology:

1. Calculate the number of pallets per day *t* that will be consumed by each lot (q_{it}) in function of the demand. See the following example based on the same data of the Database 1 (*see page 40*):

t	1	2	3	4	5	6
q_{it}	37	0	0	0	0	0
d_{it}	0	7	5	10	5	10

Table 12. *Example of incoming and demand for T=6.*

$$d_{1t} = [0,7,5,10,5,10]$$

2. **Calculation of the indicator,** this is the key element of the model to find out the best possible solution in terms of space losses. There has been stablished six alternative formulas to calculate the indicator, all related with the space losses and space utilisation:

Indicator	Formula	Units
I1	Honeycomb losses (Horizontal) + Aisle losses	m^2
I2	Honeycomb losses (Horizontal + Vertical) + Aisle losses	m^2
I3	Honeycomb losses (Horizontal) + Aisle losses + Structure space	m^2
I4	Honeycomb losses (Horizontal + Vertical) + Aisle losses + Structure space	m^2
I5	Aisle losses + Structure space	m^2
I6	Item volume / (Structure volume + Aisle volume losses)	%

Table 13. *Indicators of the model.*

The performance of the six indicators will be tested in the *Chapter 5: Validation of the Model*. All of them are related to the space losses and space utilisation due to the storing process. Below, it is showed the formulas to calculate the indicators of *Table 13.*

- ***Honeycomb losses (horizontal)***

$$Honeycomb\ losses\ (horizontal) = \left(k - c_{q_{it}itmdk}\right) \cdot (r_l \cdot r_d) \tag{22}$$

- ***Honeycomb losses (vertical)***

$$Honeycomb\ losses\ (vertical) = \frac{b_{q_{it}itmdk} - c_{q_{it}itmdk} \cdot z}{z} \cdot (r_l \cdot r_d) \tag{23}$$

- ***Aisle losses***

$$Aisle\ losses\ = (l_{itsd} + l_{itmd}) \cdot \left(\frac{a}{2}\right) \cdot (r_l) \tag{24}$$

- ***Structure space***

$$Structure\ space = \left(l_{itsd} + \sum_{k=1}^{K} l_{itmdk} \cdot k\right) \cdot (r_l \cdot r_d) \tag{25}$$

- ***Item volume***

$$Item\ volume\ = s_{it} \cdot (r_l \cdot r_d \cdot r_h) \tag{26}$$

- ***Structure volume***

$$Structure\ volume\ = \left(l_{itsd} + \sum_{k=1}^{K} l_{itmdk} \cdot k\right) \cdot z \cdot (r_l \cdot r_d \cdot r_h) \tag{27}$$

- ***Aisle volume losses***

$$Aisle\ volume\ losses\ = (l_{itsd} + l_{itmd}) \cdot \left(\frac{a}{2}\right) \cdot (r_l \cdot r_h) \tag{28}$$

To understand easily how work the parameters of the indicators see at *Table 14.*

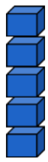
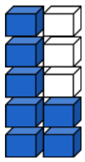
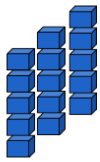
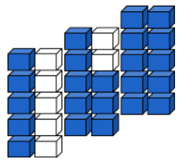
	Single deep (k=1) 1 lane	Single deep (k=1) 2 lane	Single deep (k=3) 1 lane	Single deep (k=3) 2 lanes
Distribution				
Honeycomb Losses (Horizontal)	$0 \cdot (rl * rd)$	$0 \cdot (rl * rd)$	$0 \cdot (rl * rd)$	$1 \cdot (rl * rd)$
Honeycomb Losses (Vertical)	$0 \cdot (rl \cdot rd)$	$0 \cdot (rl \cdot rd)$	$0 \cdot (rl \cdot rd)$	$0{,}4 \cdot (rl \cdot rd)$
Aisle losses	$1 \cdot \left(\frac{a}{2}\right) * rl$	$2 \cdot \left(\frac{a}{2}\right) \cdot rl$	$1 \cdot \left(\frac{a}{2}\right) \cdot rl$	$2 \cdot \left(\frac{a}{2}\right) \cdot rl$
Structure space	$1 \cdot (rl \cdot rd)$	$2 \cdot (rl \cdot rd)$	$3 \cdot (rl \cdot rd)$	$6 \cdot (rl \cdot rd)$
Item volume	$5 \cdot (rl \cdot rh \cdot rd)$	$7 \cdot (rl \cdot rh \cdot rd)$	$15 \cdot (rl \cdot rh \cdot rd)$	$23 \cdot (rl \cdot rh \cdot rd)$
Structure volume	$5 \cdot (rl \cdot rh \cdot rd)$	$10 \cdot (rl \cdot rh \cdot rd)$	$15 \cdot (rl \cdot rh \cdot rd)$	$30 \cdot (rl \cdot rh \cdot rd)$
Aisle volume losses	$1 \cdot \left(\frac{a}{2}\right) \cdot (rl \cdot rh)$	$2 \cdot \left(\frac{a}{2}\right) \cdot (rl \cdot rh)$	$1 \cdot \left(\frac{a}{2}\right) \cdot (rl \cdot rh)$	$2 \cdot \left(\frac{a}{2}\right) \cdot (rl \cdot rh)$

Table 14*. Example of the parameters of the indicators without the rack size calculation (rack length = rl; rack depth = rd; rack height = rh).*

As it has been explained, the most important is to calculate the effect of space losses along all the life cycle of a batch q_{it} in the warehouse. Hereafter, there is an example of the performance of the indicator in the **Incoming Distribution** process (see the mathematical procedure in the appendix *A.1. Incoming Distribution*):

Database 1	
Description	**Data**
Lane depth (k)	3
Maximum lanes sd	8
q_{it}	37
a	3
z	5
r_l	1,65
r_d	0,825
r_h	0,9

Table 15. *Example of database.*

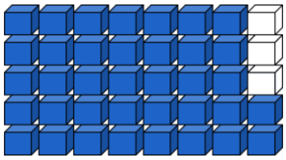
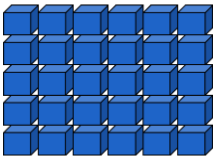
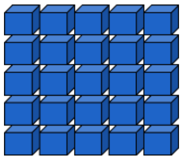
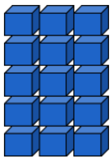
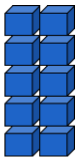
1		
t (d_{it})	**1 (0)**	
Distribution		
Honeycomb losses (horizontal)	0	
Honeycomb losses (vertical)	0	
Aisle losses	8 · 1,5 · 1,65 = 19,80	
Structure space	8 ·5 · (1,65 · 0,825) = 10,89	
Item volume	37 · (1,65 · 0,825 · 0,9) = 45,33	
Structure volume	40 · (1,65 · 0,825 · 0,9) = 49,01	
Aisle volume losses	8 · 5 · 1,5 · (1,65· 0,9) = 89,1	
t (d_{it})	**2 (7)**	**3 (5)**
Distribution		
Honeycomb losses (horizontal)	0	0
Honeycomb losses (vertical)	0	0
Aisle losses	6 · 1,5 · 1,65 = 14,85	5 · 1,5 · 1,65 = 12,375
Structure space	6 · (1,65 · 0,825) = 8,17	5 · (1,65 ·0,825) = 6,81
Item volume	30 · (1,65 · 0,825 · 0,9) = 36,75	25 · (1,65 · 0,825 · 0,9) = 30,63
Structure volume	6 · 5· (1,65 · 0,825 · 0,9) = 36,75	5 · 5 · (1,65 · 0,825 · 0,9) = 30,63
Aisle volume losses	6 · 5 · 1,5 · (1,65 · 0,9) = 66,83	5 · 5 · 1,5 · (0,65 · 0,9) = 24,38
t (d_{it})	**4 (10)**	**5 (5)**
Distribution		
Honeycomb losses (horizontal)	0	0
Honeycomb losses (vertical)	0	0
Aisle losses	3 · 1,5 · 1,65 = 7,43	2 · 1,5 · 1,65 = 4,95
Structure space	3 · (1,65 · 0,825) = 4,08	2 · (1,65 · 0,825) = 2,97
Item volume	15 · (1,65 · 0,825 · 0,9) = 18,38	10 · (1,65 · 0,825 · 0,9) = 2,45
Structure volume	3 · 5 · (1,65 · 0,825 · 0,9) = 18,38	2 · 5 · (1,65 · 0,825 · 0,9) = 2,45
Aisle volume losses	3 · 5 · 1,5 · (1,65 · 0,9) = 33,41	2 · 5 ·1,5 · (1,65 · 0,9) = 22,28

Table 16. *Example of the calculation of the indicator in the Incoming Distribution (1).*

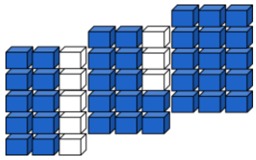
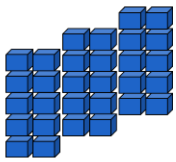
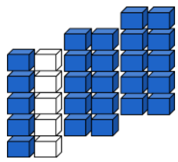
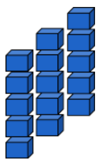
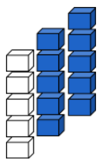
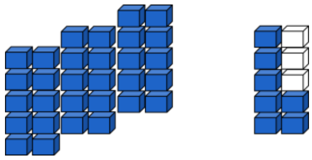
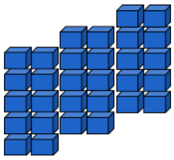
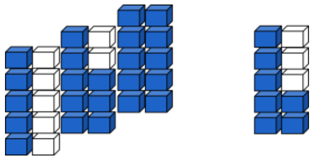
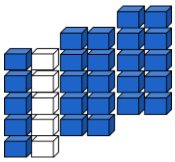
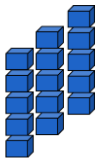
Indicator_md_tot_1		
t (d_{it})	**1 (0)**	
Distribution		
Honeycomb losses (horizontal)	1 · 1,65 · 0,825 = 1,36	
Honeycomb losses (vertical)	(3/5) · 1,65 · 0,825 = 0,82	
Aisle losses	3 · 1,5 · 1,65 = 7,43	
Structure space	3 · 3 · 1,65 · 0,825 = 12,25	
Item volume	37 · 1,65 · 0,825 · 0,9 = 45,33	
Structure volume	3 ·3 · 5 · (1,65 · 0,825 · 0,9) = 55,13	
Aisle volume losses	3 · 5 · 1,5 · 1,65 · 0,9 = 33,41	
t (d_{it})	**2 (7)**	**3 (5)**
Distribution		
Honeycomb losses (horizontal)	0	1 · 1,65 · 0,825 = 1,36
Honeycomb losses (vertical)	0	0
Aisle losses	2 · 1,5 · 1,65 = 4,95	2 · 1,5 · 1,65 = 4,95
Structure space	2 · 3 · (1,65 · 0,825) = 8,17	2 · 3 · (1,65 · 0,825) = 8,17
Item volume	30 · (1,65 · 0,825 · 0,9) = 36,75	25 · (1,65 · 0,825 · 0,9) = 30,63
Structure volume	2 · 3 · 5 · (1,65 · 0,825 · 0,9) = 36,75	2 · 3 · 5 · (1,65 · 0,825 · 0,9) = 36,75
Aisle volume losses	2 · 5 ·1,5 · (1,65 · 0,9) = 22,28	2 · 5 ·1,5 · (1,65 · 0,9) = 22,28
t (d_{it})	**4 (10)**	**5 (5)**
Distribution		
Honeycomb losses (horizontal)	0	1 · 1,65 · 0,825 = 1,36
Honeycomb losses (vertical)	0	0
Aisle losses	1 · 1,5 · 1,65 = 2,48	1 · 1,5 · 1,65 = 2,48
Structure space	1 · 3 · (1,65 · 0,825) = 4,08	1 · 3 · (1,65 · 0,825) = 4,08
Item volume	15 · (1,65 · 0,825 · 0,9) = 18,38	10 · (1,65 · 0,825 · 0,9) = 12,25
Structure volume	1 · 3 · 5 · (1,65 · 0,825 · 0,9) = 18,38	1 · 3 · 5 · (1,65 · 0,825 · 0,9) = 18,38
Aisle volume losses	1· 1,5 · (1,65 · 0,9) = 2,23	1 · 1,5· (1,65 · 0,9) = 2,23

Table 17. *Example of the calculation of the indicator in the Incoming Distribution (2).*

Indicator_md_tot_2		
t (d_{it})	**1 (0)**	
Distribution		
Honeycomb losses (horizontal)	0	
Honeycomb losses (vertical)	0	
Aisle losses	(2 + 2) · 1,5 · 1,65 = 9,90	
Structure space	((2· 3) + 2) · (1,65 · 0,825) = 10,89	
Item volume	37· (1,65 · 0,825 · 0,9) = 45,33	
Structure volume	((2· 3) + 2) · 5 · (1,65 · 0,825 · 0,9) = 49,01	
Aisle volume losses	(2 + 2) · 5 · 1,5 · (1,65 · 0,9) = 44,55	
T (d_{it})	**2 (7)**	**2 (7)**
Distribution		
Honeycomb losses (horizontal)	0	1· (1,65 · 0,825) = 1,36
Honeycomb losses (vertical)	0	(2/5) · (1,65 · 0,825) = 0,54
Aisle losses	2 · 1,5 · 1,65 = 4,95	(2 + 2) · 1,5 · 1,65 = 9,9
Structure space	2 · 3 · (1,65 · 0,825) = 8,17	((2 · 3) + 2) · (1,65 ·0,825) = 10,89
Item volume	30 · (1,65 · 0,825 · 0,9) = 36,75	30 · (1,65 · 0,825 · 0,9) = 36,75
Structure volume	2 · 3 · 5 · (1,65 · 0,825 · 0,9) = 36,75	((2 · 3 ·5) + (2 · 5)) · (1,65 · 0,825 · 0,9) = 49,01
Aisle volume losses	2 · 5 · 1,5· (1,65 · 0,9) = 22,28	((2 · 5) + (2 · 5)) · (1,65 · 0,9) = 29,70
t (d_{it})	**3 (5)**	**4 (10)**
Distribution		
Honeycomb losses (horizontal)	1 · (1,65 · 0,825) = 1,36	0
Honeycomb losses (vertical)	0	0
Aisle losses	2 · 1,5 · 1,65 = 4,95	1 · 1,5 · 1,65 = 2,48
Structure space	2 · 3 · 1,65 · 0,825 = 8,17	1 · 3 · 1,65 · 0,825 = 4,08
Item volume	25 · (1,65 · 0,825 · 0,9) = 30,63	15 · (1,65 · 0,825 · 0,9) = 18,38
Structure volume	2 · 3 · 5 · (1,65 · 0,825 · 0,9) = 36,75	1 · 3 · 5 · (1,65 · 0,825 · 0,9) = 18,38
Aisle volume losses	2 · 5 ·1,5 · (1,65 · 0,9) = 22,28	1 · 5 · 1,5 · (1,65 · 0,9) = 11,14

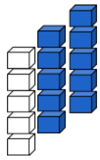
t (d_{it})	5 (5)
Distribution	
Honeycomb losses (horizontal)	1 · 1,65 · 0,825 = 1,36
Honeycomb losses (vertical)	0
Aisle losses	1 · 1,5 · 1,65 = 2,48
Structure space	1 · 3 · 1,65 · 0,825 = 9,03
Item volume	10 · (1,65 · 0,825 · 0,9) = 12,25
Structure volume	1 · 3 · 5 · (1,65 · 0,825 · 0,9) = 18,38
Aisle volume losses	1 · 5 · (1,5 · 1,65 · 0,9) = 11,14

Table 18. *Example of the calculation of the indicator in the Incoming Distribution (4).*

Indicator_sd_tot_1						
Indicator	**t = 1**	**t = 2**	**t = 3**	**t = 4**	**t = 5**	**Total**
Indicator 1	19,85	14,85	12,38	7,43	4,95	**59,46**
Indicator 2	19,85	14,85	12,38	7,43	4,95	**59,46**
Indicator 3	30,74	23,02	19,19	11,51	7,92	**92,38**
Indicator 4	30,74	23,02	19,19	11,51	7,92	**92,38**
Indicator 5	30,74	23,02	19,19	11,51	7,92	**92,38**
Indicator 6	0,33	0,35	0,56	0,35	0,10	**1,69**

Table 19. *Results of the calculation of the Indicator_sd_tot_1.*

Indicator_md_tot_1						
Indicator	**t = 1**	**t = 2**	**t = 3**	**t = 4**	**t = 5**	**Total**
Indicator 1	8,79	4,95	6,31	2,48	3,84	**26,37**
Indicator 2	9,61	4,95	6,31	2,48	3,84	**27,19**
Indicator 3	21,04	13,20	14,48	6,56	7,92	**63,2**
Indicator 4	21,86	13,12	14,48	6,56	7,92	**63,94**
Indicator 5	19,68	13,12	13,12	6,56	6,56	**59,04**
Indicator 6	0,51	0,62	0,52	0,89	0,59	**3,13**

Table 20. *Results of the calculation of the Indicator_md_tot_1.*

Indicator_md_tot_2							
Indicator	**t = 1**	**t = 2**		**t = 3**	**t = 4**	**t = 5**	**Total**
Indicator 1	9,9	4,95	11,26	6,31	2,48	3,84	**27,48**
Indicator 2	9,9	4,95	11,80	6,31	2,48	3,84	**27,48**
Indicator 3	20,79	13,12	22,15	14,48	6,56	12,87	**67,82**
Indicator 4	20,79	13,12	22,69	14,48	6,56	12,87	**67,82**
Indicator 5	20,79	13,12	20,79	13,12	6,56	11,51	**65,10**
Indicator 6	0,48	0,62	0,47	0,52	0,62	0,41	**3,12**

Table 21. *Results of the calculation of the Indicator md_tot_2.*

In the case 4 (*Indicator md_tot_2*) it can be seen that there is one moment (*t = 2*) that the program has to decide which emptying movement would be the most suitable one in terms of space efficiency. In *t = 2* the program calculates both possibilities and choose the one with lowest indicator value. On one hand, there is the possibility to empty 7 pallets from the single-deep lanes and the indicator value would be 3 (Aisle losses = 3), on the other hand there is the possibility to empty the 7 pallets from the multi-deep lane (k=3) and the indicator value would be 7 (Aisle losses = 6 and Honeycomb losses = 1). Following the stablished methodology, the model chooses the first option and continues the iteration process with this decision till the end of the calculation of the total indicator.

Database 1 (k=3; max_lanes_sd = 8)						
Description	**I1**	**I2**	**I3**	**I4**	**I5**	**I6**
Indicator_sd_tot_1	59,46	59,46	92,38	92,38	92,38	1,69
Indicator_sd_tot_2	N/A	N/A	N/A	N/A	N/A	N/A
Indicator_md_tot_1	26,37	27,19	63,2	63,94	59,04	3,13
Indicator_md_tot_2	N/A	N/A	N/A	N/A	N/A	N/A
Indicator_md_tot_3	27,48	27,48	67,82	67,82	65,10	3,12

Table 22. *Results summary of the calculation of the indicator.*

In this case, with the specific parameters of k-depth=3 and 8 single-deep lanes available the model would assign the batch q_{it} to multi-deep racks instead of single-deep or a combination of both of them. The model will keep the value of the indicator and will repeat the same process for all the possible combinations of single-deep and different k-depth lanes. The combination that gets the lowest indicator will be the selected by the model and the batch q_{it} will be stored in this rack. When a batch q_{it} is located to one spot, this product cannot be moved to any other lane even if it is the same product *i*.

Also, in *Table 22* it can be seen that with these parameters, the decision is the same for the six indicators. It doesn't mean that at the end of the iterations the final decision will be the same because even though the results that the algorithm get with the indicators are similar, probably with different parameters the decision for the different indicators will be different. By this way the final result of the model may be similar but probably different. In chapter 5. *Validation of the model*, there are four different experiments to study which is the best indicator depending on the demand, production and stock.

The performance of the indicator in the inventory control is exactly the same as the Incoming distribution but without the iteration till the complete consumption of the batch. Once you assign on batch to a particular lane, you are assuming how will be its consumption because it has already been calculated.

After applying the Local Optimization process to all the different products *i*, the output of this model is a solution for each specific product of the combination of k-deep lanes for each day *t* and each product *i*.

	1	2	3	4	5
i= 1	2 of k=2	1 of k =2 3 of k = 3	1 of k =1 3 of k = 3	1 of k =1 2 of k = 3	1 of k =1 1 of k =3
i = 2	1 of k=8	1 of k=8	1 of k=1 1 of k =8	2 of k = 2 1 of k = 8	2 of k = 2
i = 3	1 of k = 1	1 of k=1 3 of k = 9	2 of k=9	2 of k =2 1 of k =9	2 of k = 2

Table 23. *Fictional results after the Local Optimization process.*

4.4.2. Relocation

i. Description

In the first step of the model, it achieves the best local solutions. For each product *i* and each day *t*, it gets a different number of k-deep lanes based on its incoming and its demand. To export the solution to a real warehouse, it will be necessary to choose a specific number of different areas. It makes much more sense and is more much realistic thinking in the organisation and management of a warehouse as sample of just a few number of different depths.

Therefore, the objective of "The Relocation" process is to relocate the articles to a specific depth *k* even though it is not the optimal solution individually. First of all, it is required to count the number of each lane depth during all the horizon time taken into consideration during the analysis and select the final depth *k* candidates of the warehouse (*k-list*). Then, the process performs the same process as "Local Optimization" but instead of trying all possible k (*k = 2,..,max_k_depth*), it iterates through the specific *k* (k-list) most used in the solution of "Local Optimization". By using this strategy, it will be guaranteed that the maximum possible number of products keep their optimal solution.

ii. Input / Output

Figure 18. *Input and Output of the Relocation process.*

iii. Procedure

To carry out this process each article will perform the next steps:

Step 0. (Initialization) Set the number of products that have to be stored in the warehouse ($i=1,..I$), the rack dimensions (r_l, r_h y r_d), the stackability (z), the aisle width (a) and the costs variables (c_{sd}, c_{md}, c_s and c_h).

Step 1. Read the initial data of product i to determine the q_{it} (incoming of product i at day t), the q_{it} (demand of product i at day t), the horizon time ($t = 1,..,T$) and $d_{q_{it}}$ (consumption of the batch q_{it} at day t).

Step 2. Carry out the **Pre-Process** to calculate the maximum number of lanes of single deep (*maximum lanes sd*) and the list of k candidates (*k-list*).

Step 3. Initialization of the variables of the *time*, the *stock* and the *Indicator_def*: $t = 1$; $s_{sd} = 0$; $l_{sd} = 0$; $s_{md} = 0$; $l_{md} = 0$; $s_{q_{it}sd} = 0$; $l_{q_{it}sd} = 0$; $s_{q_{it}mdk} = 0$; $l_{q_{it}mdk} = 0$; $b_{q_{it}mdk} = 0$; $c_{q_{it}mdk} = 0$; $k_{q_{it}mdk} = 0$; $Indicator\_def = M$.

Step 4. If at day t the incoming of product I is bigger than 0 ($q_{it} > 0$) execute the **"Incoming Distribution"** process.

Step 5. If at day t the demand of product i is bigger than 0 ($d_{it} > 0$) execute the **"Inventory Control"** process.

Step 6. The algorithm moves to next day. Update the *time, the stock* and the *Indicator_def*: $t = t + 1$; s_{sd}; l_{sd}; s_{md}; l_{md}; $s_{q_{it}sd}$; $l_{q_{it}sd}$; $s_{q_{it}mdk}$; $l_{q_{it}mdk}$; $b_{q_{it}mdk}$; $c_{q_{it}mdk}$; $k_{q_{it}mdk}$; $Indicator\_def = M$.

Step 7. Repeat Step 1 to Step 6 for all the products. Update i: $i = i + 1$

Step 8. The process finishes when there are no more products to store in the warehouse (i=I).

Examining the algorithm deeply, the activities carried out by the algorithm are very similar to the Local Optimization process. The main difference is found in the **Pre-process,** now it does not calculate the variable *maximum k-depth*, it calculates a list of different depth candidates called *k-list* in function of the number of areas of the warehouse. This number of areas will be determined by the layout of the warehouse or for the requirements of the client. By this way, the model starts to translate from a local solution to a global solution. Hereunder, it is explained the performance of the Pre-process:

1. **Pre-process**: In addition to calculating the variable of the *maximum number of lanes of single deep*, it calculates the *k-list* to make sure the algorithm explores the most suitable depths based on the results of the Local Optimization process that principally depend on the characteristics of the incoming and the demand of all the products.

Step 2. Count the number of each lane depth during all the horizon time.

$$k_{candidate} = \sum_{i=1}^{I} \sum_{t=1}^{T} k_{q_{it}mdk} \qquad \forall\, k \tag{29}$$

Step 3. Taking into consideration the number of areas (N) with multi-deep lanes ($k > 1$), create the *k-list* with the N maximum $k_{candidate}$.

The single-deep lane ($k = 1$) is not taken into consideration to create the list because of the flexibility that they can give to a warehouse overcomes its space losses in some specific cases, and if they are not needed, in the previously steps the model should have taken them out of the configuration.

For instance, if the results of the Local Optimization process are the results of the *Table 24. Number of lanes of each k-depth per product i at day t*:

	t=1	t=2	t=3	t=4	t=5
i= 1	2 of k=2	1 of k =2 3 of k = 3	1 of k =1 3 of k = 3	1 of k =1 2 of k = 3	1 of k =1 1 of k =3
i = 2	1 of k=8	1 of k=8	1 of k=1 1 of k =8	2 of k = 2 1 of k = 8	2 of k = 2
i = 3	1 of k = 1	1 of k=1 3 of k = 9	2 of k=9	2 of k =2 1 of k =9	2 of k = 2

Table 24. *Number of lanes of each k-depth per product i at day t*

The outcome of *Step 1* would be the following one:

	k = 1	k = 2	k = 3	k= 4	k =5	k = 6	k = 7	k = 8	k = 9
Number of lanes	6	11	9	0	0	0	0	4	6

Table 25. *Total amount of lanes of each k-depth.*

If the number of areas was 2 ($N = 2$), then $k = 2$ and $k = 3$ would make up the *k-list*. If it was planned to set three different areas ($N = 3$), then $k = 2$, $k = 3$ and $k = 9$ would make up the list.

4.4.3. Redistribution

i. Description

The Redistribution process is the last one applied in the model and the main objective is to relocate some q_{it} that are assigned to a specific depth (k) to another depth (k) to balance the solution. When the model is translating the solution to a global solution there is an important gap that has to be solved, if a solution is optimum locally it doesn't mean that is optimum globally.

The number of lanes of k-depth that need the warehouse is the **maximum number** of lanes of k-depth at any day t. This means that if at day 1 is only needed 15 lanes of multi-deep k=3, but at day 10 is needed 20 lanes of multi-deep k=3, the warehouse needs 20 lanes of multi-deep k=3 even though at day 1 there are 5 lanes of multi-deep k=3 that are empty. Therefore, the redistribution process looks for a more homogeneous solution reducing the maximum number of lanes of any k-depth to reduce the surface utilization and to increase the storage efficiency of the warehouse. To clarify this issue, look at the following example of the possible results after the ·Relocation" process:

	1	2	3	4	5
k = 1	1	2	4	3	4
k = 2	5	4	4	3	3
k = 4	1	2	3	2	2

Table 26. *Fictional results of the Redistribution process (k-list)*

With the results of the *Table 26*, at least, this warehouse needs 4 lanes of single-deep racks, 5 lanes of multi-deep (k=2) racks and 3 lanes of multi-deep (k=4) racks. The strategy embraced in the Redistribution process focuses on looking for the day that is employing the maximum number of lanes and trying to relocate the products of this k-depth to another k-depth where there are available empty lanes with pallet locations. For instance, if the algorithm is studying the alternatives of products located in k=2 lanes, the first step is trying to relocate some of the pallets at day 1 to the empty single-deep, to the empty multi-deep racks (k=4) or to a combination of both of them.

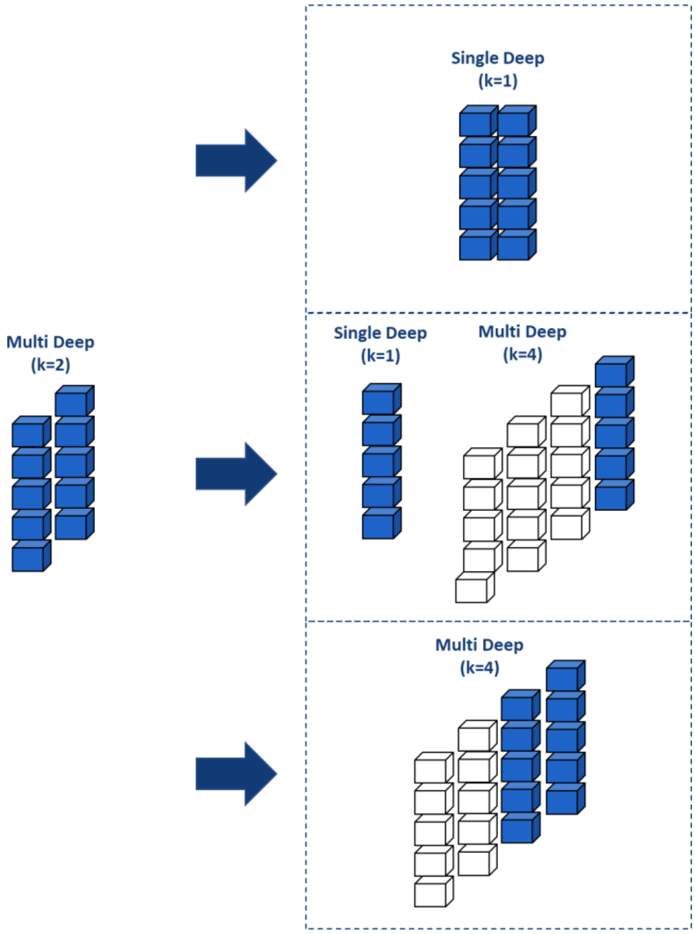


Figure 19. *Redistribution alternatives*

The **Redistribution process** follows three main steps to relocate the products, the "**Initial Peak Redistribution"** process, the **"Backward Redistribution"** process and the "**Forward Redistribution"** process. On this occasion, the approach is a substantially different from the Local Optimization process and the Relocation process. Even though the algorithm cements the performance of the three main steps relocating the products following the five different options explained in the Incoming Distribution process, now, the indicator is not the key element that induces to make the decision. In the redistribution process, the key element that induces to choose between the different alternatives is the space utilisation of the warehouse. The main objective now is to reduce the space utilisation due to the redistribution of the incoming products.

First, in the first step, the **Initial Peak Redistribution** process, the model focuses on relocating the products at the day *t* with the maximum number of lanes *k-depth*. Moreover, the Redistribution process doesn't finish with the possible relocation of the incoming products of the day with the maximum lanes, it goes through two more process exploring the possible redistribution going back in time, that is named the **Backward Redistribution** process and going forward in time, that is named the **Forward Redistribution**.

ii. Input / Output

Figure 20. *Input and Output of the Redistribution process.*

iii. Procedure

The Redistribution process is implemented with the following steps:

Step 1. (Initialisation) Read the data of the results of the "Relocation" process. Set the k-list and the inventory results of all the products: s_{sd}; l_{sd} ; s_{md}; l_{md} ; $s_{q_{it}sd}$; $l_{q_{it}sd}$; $s_{q_{it}mdk}$; $l_{q_{it}mdk}$; $b_{q_{it}mdk}$; $c_{q_{it}mdk}$; $k_{q_{it}mdk}$.

Step 2. Carry out the "**Initial Peak Redistribution"** process:

Step 2.1. Determine the day with the maximum number of multi-deep lanes of depth *k (k > 1)*. Set the variables of the *depth* and *time*: *k* and *t*.

Set the variable *product*: *i = 1*.

Step 2.2.

If product *i* at day *t* has incoming (q_{it}) and if incoming (q_{it}) is located in a *k* rack execute the **"Redistribution Iteration"** process.

Step 2.2.1.

Relocate incoming (q_{it}) to **single-deep** lane following the **case 1** of the **Incoming Distribution** process.

Calculate the new Space utilization (Space_utilisation_2.2.1 = Space utilisation)

Step 2.2.2.

Relocate incoming (q_{it}) to **single-deep** and **multi-deep** lanes of depth *k* different than the *k* studied following the **case 2** of the **Incoming Distribution** process.

Calculate the new Space utilization (Space_utilisation_2.2.2 = Space utilisation).

Repeat *Step 2.2.3.* until there are no other possibilities to explore (All depth *k* options in *k-list*).

Step 2.2.3.

Relocate incoming (q_{it}) to **multi-deep** lanes of depth k different than the studied k following the **case 3** of the **Incoming Distribution** process.

Calculate the new space utilization (Space_utilisation_2.2.3 = Space utilisation).

Repeat *Step 2.2.3.* until there are no other possibilities to explore (All depth k options in k-list).

Step 2.2.4.

Relocate incoming (q_{it}) to **multi deep** lanes of depth k different than the studied k and ***single-deep*** lanes following the **case 4** of the **Incoming Distribution** process.

Calculate the new space utilization ((Space_utilisation_2.2.4 = Space utilisation).

Repeat *Step 2.2.4.* until there are no other possibilities to explore (All depth k in *k-list*).

Step 2.2.5.

Relocate incoming (q_{it}) to **multi deep** lanes of depth k different than the studied k following the **case 5** of the **Incoming Distribution** process.

Calculate the new space utilization (m^2).

Repeat Step 2.2.5. until there are no other possibilities to explore (All depth k options in *k-list*).

If the new space utilisation (m^2) is smaller than the current space utilisation: Select the new space utilisation (m^2) as the current space utilisation (m^2) and update the k-list and the inventory variables (Output = New space utilisation).

Update the product: $i = i + 1$

Repeat Step 2.2. for all the products, until there are no more possibilities to explore ($i = I$).

Repeat Step 2.1. until there are not any more depth k to explore in the *k-list*.

Step 3. Repeat Step 2 while the Space utilisation calculated at the end of Step 2 is smaller than at the beginning (*Output < Input*).

Step 4. It starts the **"Backward redistribution"** process: Determine the maximum number of multi-deep lanes of depth k of the k-list. It starts with the first depth k of multi-deep lanes and saves the depth k and day t with this maximum number of lanes.

Set the variable of the *time* and the *backward position* (var): *var; t = day t with the maximum number of multi-deep lanes of depth k – var.*

Step 4.1.

Execute *Step 2.1.* for all the products.

Update the variable of the backward *position*: *var = var – 1*.

Step 4.2.

Set the variable of the *time*: *t = day t* with the maximum number of multi-deep lanes of depth *k.*

Execute Step 2.

Step 5. Repeat *Step 4* until the variable of the *time* in *Step 4.1.* is smaller than 0 ($t < 0$) in all depths *k.*

Step 6. It starts the **"Forward redistribution"** process: Determine the maximum number of multi-deep lanes of depth *k* of the k-list. It starts with the first depth *k* of multi-deep racks and saves the depth *k* and day *t* with this maximum number of lanes.

Set the variable of the *time* and the *forward position* (var): *var; t = day t with the maximum number of multi-deep lanes of depth k + var*;

Step 6.1.

Execute *Step 2.1.* for all the products.

Update the variable of the *forward position*: *var = var + 1*.

Step 6.2.

Set the variable of the time *t = day t with the maximum number of multi-deep lanes of depth k.*

Execute *Step 2* for all the products.

Step 7. Repeat Step 6 until the variable *t* in Step 6.1. is bigger than *T* ($t > T$) in all depths *k*.

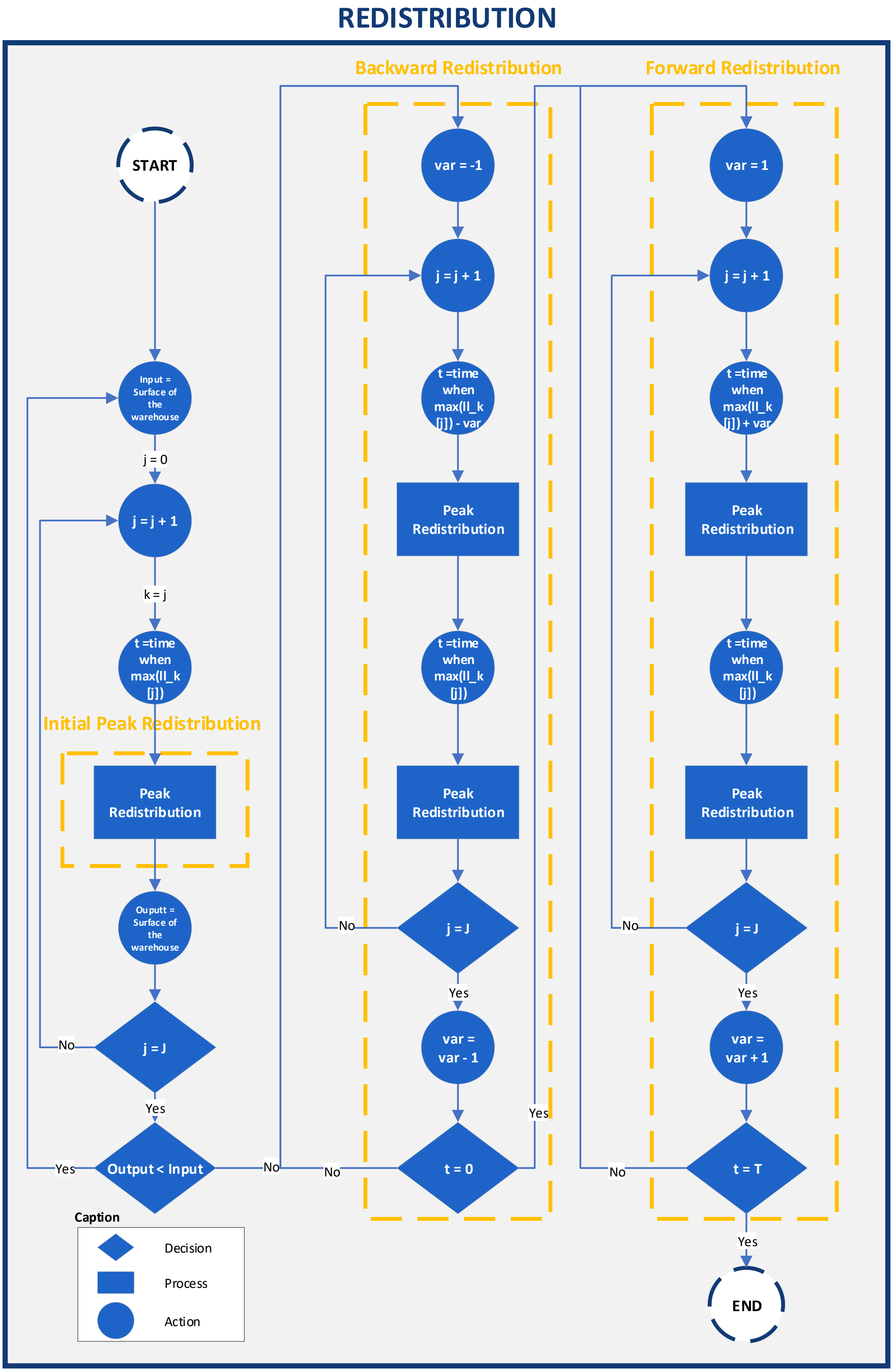


Figure 21. Redistribution Process

PEAK REDISTRIBUTION

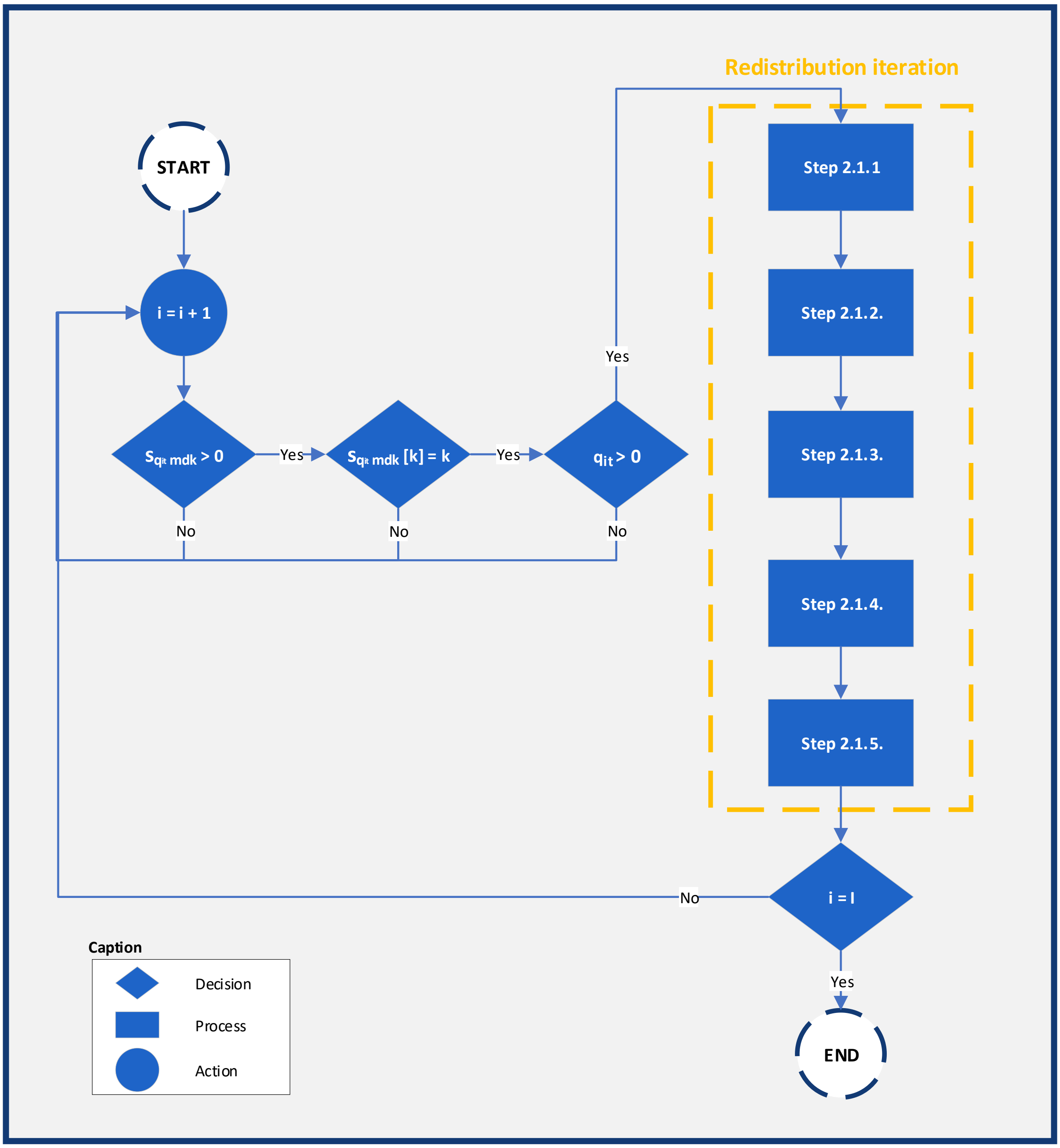


Figure 22. Peak Redistribution Process

5. Validation of the model

The aim of this chapter is to validate if the model created in this Master Thesis works for any kind of database and to find out which is the best indicator in terms of final results, if there is one indicator that performs better for any kind of warehouse inventory rotation or instead there are some specific indicators for each kind of data.

To test the model, four experiments are carried out, with different types of database that variates the amount and the frequency of incoming with a similar demand. It is thus easier to test four types of warehouse with the following features:

- **Experiment 1:** A low volume of stock and a low rotation of stock (*5.3.1*).
- **Experiment 2:** A low volume of stock and a high rotation of stock *(0).*
- **Experiment 3:** A high volume of stock and a low rotation *(0).*
- **Experiment 4:** A high volume of stock and a high rotation *(0).*

5.1. Datasets

Initial Stock: The calculation of the inventory level in the first period should be calculated based on the stock level of the warehouse. In the implementation of the model the first incoming at t=1 is assumed the initial stock. Thus, there is no need for a specific variable for the initial stock and the initial stock is then part of the incoming data.

Stock breakdown: The model is not able to stand stock breakdown situations; in all the horizon time t the incoming production has to make sure that all the demand can be supplied by the incoming production.

5.1.1. Input Data

Database			
Description	**Data**	**Description**	**Data**
a	3	Cost single-deep rack	30 €/pallet location
z	5	Cost multi-deep rack	70 €/pallet location
r_l	1,65 m	Cost m^2	500 €
r_d	0,825 m	Cost lift truck	45.000 €
r_h	0,9 m	Products	20
Areas	3	Time	40

Table 27. *Database of the four experiments*

To see the incoming and the demand go to Appendix B.1. and to see the inventory level through all the horizon time, go to Appendix B.2.

5.1.2. Output Data

The definition and the calculation of the Output Data of the Model is determined in Chapter *4.4 Multi-deep storage system model*. The results of the Experiments will be showed with the format of the following table:

Output Data	
Description	**Units**
Surface of the warehouse	m^2
Storage capacity	p (pallet location)
Storage efficiency	% (pallets occupied/pallet locations)
Surface efficiency	p (pallet location)/m^2
Cost of the warehouse	€
Cost per net pallet location	€/p (pallet location)

Table 28. *Output data of the Experiments*

To see the evolution of the maximum number of single-deep and multi-deep lanes along the main process of the Model look at the *Appendix B.3. Results of the maximum number of single-deep and multi-deep lanes*.

5.2. Programming

The Optimization Model for multi-deep storage systems has been implemented in Python using Python's Integrated Development and Learning Environment. The program is run on an Intel® Core™ i5-5200U 2,20 GHz CPU with 8 GB RAM memory.

As programming language Python is chosen because it is an interpreted programming language whose philosophy emphasizes a syntax that favors a readable code. It is a multi-paradigm programming language, since it supports object orientation, imperative programming and, to a lesser extent, functional programming. It is strong and fast enough to execute the code in a reasonable time, yet simple enough to limit exhaustive programming.

5.3. Results of the experiments

5.3.1. Experiment with a low volume of stock and a low rotation of stock

	Experiment 1 Results					
	I1	I2	I3	I4	I5	I6
Local Optimization	224,61 m²	256,53 m²	266,06 m²	244,53 m²	236,12 m²	373,97 m²
	425 p	415 p	450 p	380 p	440 p	710 p
	36,02 %	36,89 %	34,02 %	40,28 %	34,79 %	22,01 %
	0,68 p/m²	0,60 p/m²	0,56 p/m²	0,63 p/m²	p/m²	0,42 p/m²
	184.653,13 €	194.716,87 €	205.643,75 €	185.865,00 €	190.857,5 €	274.686,25€
	434,48 €/p	469,20 €/p	456,99 €/p	489,12 €/p	433,77 €/p	386,88 €/p
Relocation	234, 51 m²	253,81 m²	273,49 m²	256,04 m²	247,13 m²	412,83 m²
	355 p	405 p	450 p	395 p	435 p	880 p
	43,12 %	37,80 %	34,02 %	38,75 %	35,18 %	17,39 %
	0,76 p/m²	0,60 p/m²	0,56 p/m²	0,60 p/m²	0,62 p/m²	0,37 p/m²
	165.949,37 €	192.255,63 €	205.643,75 €	192.069,38 €	193.814,38 €	306.215,00 €
	467,46 €/p	474,71 €/p	456,986 €/p	486,25 €/p	445,55 €/p	347,97 €/p
Redistribution (Initial Peak Redistribution)	200,60 m²	212,23 m²	222,63 m²	224,85 m²	215,70 m²	359,99 m²
	355 p	325 p	345 p	335 p	365 p	795 p
	43,12 %	47,60 %	45,20 %	46,07 %	42,14 %	19,54 %
	0,76 p/m²	0,73 p/m²	0,70 p/m²	0,69 p/m²	0,71 p/m²	0,43 p/m²
	165.949,37€	165.865,63 €	172.663,13 €	172.276,88 €	172.598,13 €	276.044,38 €
	467,46 €/p	510,36€ €/p	500,47 €/p	514,26 €/p	472,87 €/p	347,23 €/p
Redistribution (Backward Redistribution)	195,15 m²	212,23 m²	213, 59 m²	220,77 m²	195,77 m²	328,31 m²
	335 p	325 p	330 p	320 p	310 p	715 p
	45,69 %	47,60 %	46,95 %	48,23 %	49,70 %	22,30 %
	0,78 p/m²	0,73 p/m²	0,73 p/m²	0,70 p/m²	0,79 p/m²	0,49 p/m²
	161.226,88 €	165.865,63 €	167.296,25 €	168.585,00 €	158.386,25 €	254.604,38 €
	481,27 €/p	510,36 €/p	506,96 €/p	526,83 €/p	510, 92€/p	356,09 €/p
Redistribution (Forward Redistribution)	195,15 m²	212,23 m²	213, 59 m²	220,77 m²	195,77 m²	328,31 m²
	335 p	325 p	330 p	320 p	310 p	715 p
	45,69 %	47,60 %	46,95 %	48,23 %	49,70 %	22,30 %
	0,78 p/m²	0,73 p/m²	0,73 p/m²	0,70 p/m²	0,79 p/m²	0,49 p/m²
	161.226,88 €	165.865,63 €	167.296,25 €	168.585,00 €	158.386,25 €	254.604,38 €
	481,27 €/p	510,36 €/p	506,96 €/p	526,83 €/p	510, 92€/p	356,09 €/p

Table 29. *Results of the experiment with a low volume of stock and a low volume of rotation.*

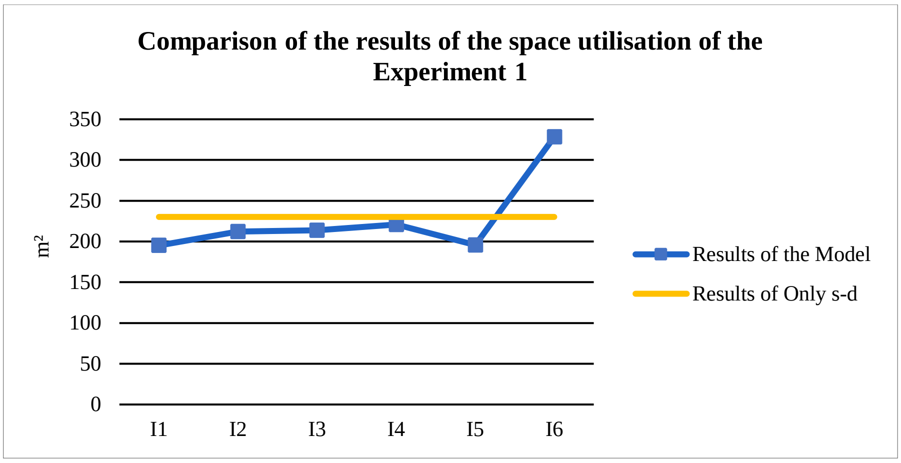


Figure 23. *Comparison of the* ***space utilisation*** *of the experiment with a low volume of stock and a low rotation.*

In this scenario, there are two indicators that perform much better than the others, these are indicator 1 and Indicator 5. Moreover, Indicator 6 performs an unsatisfactory outcome in comparison with all the other indicators. The following table shows the comparison between the output results of the Model in relation with a storage system of only single-deep lanes.

Comparison of the best results of the Experiment 1		
Results I1	Only single deep	Improvement I1
195,15 m^2	230,18 m^2	15,22 %
335 p	300 p	-10,45 %
45,69 %	51,03 %	-10,46 %
0,78 p/m^2	0,67 p/m^2	14,10 %
161.226,88 €	161.887,5 €	0,41%
481,27 €/p	539,63 €/p	10,81%

Table 30. *Comparison of the best results of the experiment with a low volume of stock and a low rotation of stock with an only single-deep storage system.*

As can be seen in *Table 10*, the Optimization Model is able to ameliorate the space utilisation of a single-deep storage system by approximately 15%, the surface efficiency over 15% and the cost around 10%. Nevertheless, the storage efficiency diminishes.

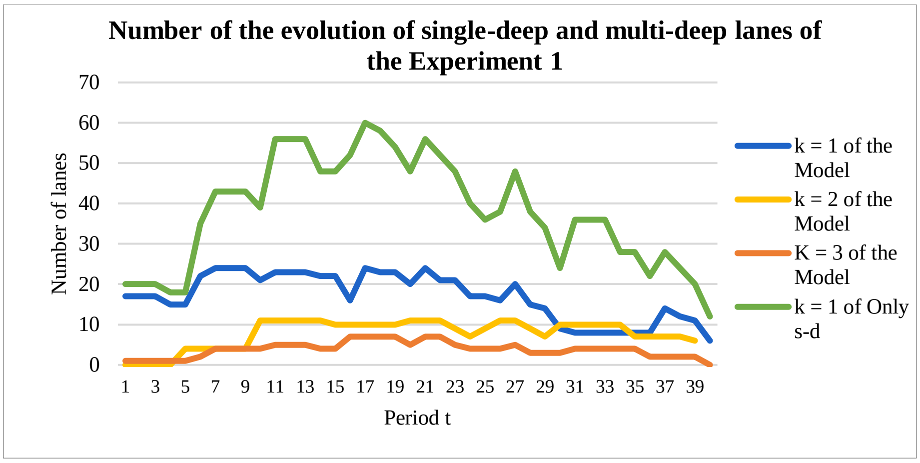


*Figure 24.*Comparison of the **evolution of the number** of lanes of single-deep and multi-deep of the experiment with a low volume of stock and a low rotation of stock.

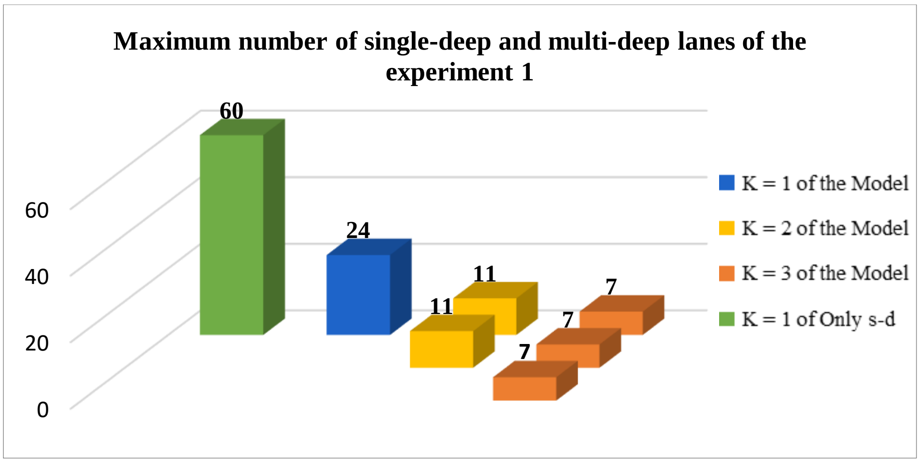


Figure 25. Comparison of the m**aximum number** of single-deep and multi-deep lanes of the experiment with a low volume of stock and a low rotation of stock.

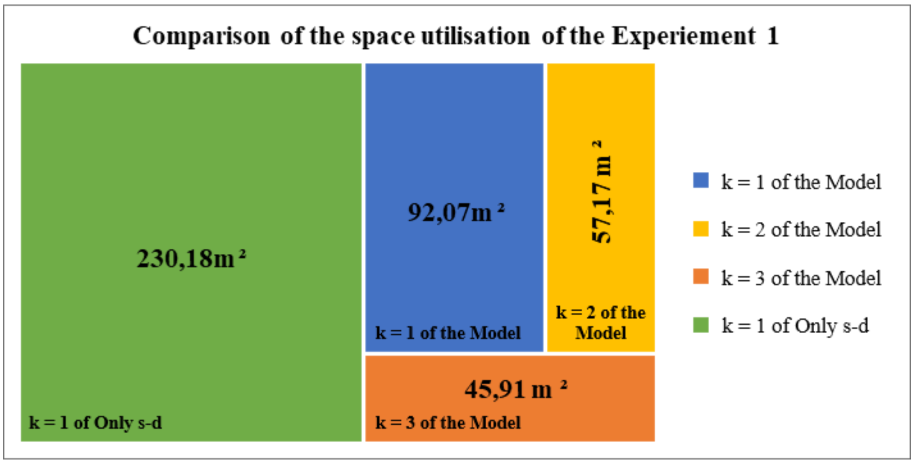


Figure 26. Comparison of the **space utilisation** of the experiment with a low volume of stock and a low rotation of stock.

5.3.2. Experiment with a low volume of stock and a high rotation of stock

	Experiment 3 results					
	I1	I2	I3	I4	I5	I6
Local Optimization	574,32 m^2	578,28 m^2	573,33 m^2	605,26 m^2	605,76 m^2	757,23 m^2
	1055 p	1.015 p	1.015 p	1.005 p	1.125 p	1.445 p
	29,73 %	30,70 %	30,70 %	31,19 %	27,88 %	29,58 %
	0,55 p/m^2	0,54 p/m^2	0,54 p/m^2	0,52 p/m^2	0,52 p/m^2	0,56 p/m^2
	399.211,87 €	394.991,87 €	392.716,87 €	403.780,62 €	419.628,13 €	512.563,12 €
	378,40 €/p	389,15 €/p	386,91 €/p	401,77 €/p	373,00 €/p	354,71 €/p
Relocation	547,84 m^2	555,64 m^2	551,80 m^2	595,24 m^2	561,21 m^2	1004,35 m^2
	985 p	950 p	945 p	950 p	1.025 p	2.380 p
	31,85 %	32,80 %	32,98 %	33,00 %	30,40 %	13,81 %
	0,57 p/m^2	0,56 p/m^2	0,56 p/m^2	0,53 p/m^2	0,56 p/m^2	0,33 p/m^2
	379.470,62 €	377.318,75 €	374.650,62 €	391.918,75 €	388.953,13 €	700.977,50 €
	385,25 €/p	397,18 €/p	396,46 €/p	412,55 €/p	379,47 €/p	294,53 €/p
Redistribution (Initial Peak Redistribution)	533,36 m^2	517,89 m^2	508,61 m^2	533,61 m^2	508,74 m^2	925,53 m^2
	950 p	875 p	850 p	860 p	905 p	2.145 p
	33,81 %	37,50 %	39,49 %	40,34%	36,89 %	18,03 %
	0,60 p/m^2	0,63 p/m^2	0,66 p/m^2	0,65 p/m^2	0,66 p/m^2	0,42 p/m^2
	369.181,25 €	352.996,88 €	345.606,25 €	357.805,00 €	354.918,13 €	645.313,13 €
	388,61 €/p	403,43 €/p	406,60 €/p	416,05 €/p	392,17 €/p	300,85 €/p
Redistribution (Backward Redistribution)	474,83 m^2	466,41 m^2	464,56 m^2	499,08 m^2	454,04 m^2	887,29 m^2
	835 p	795 p	770 p	815 p	795 p	2.050 p
	37,57 %	42,59 %	42,29 %	43,99 %	41,07 %	16,03 %
	0,66 p/m^2	0,73 p/m^2	0,70 p/m^2	0,72 p/m^2	0,72 p/m^2	0,47 p/m^2
	331.864,38 €	324.856,88 €	320.178,75 €	338.991,88 €	320.069,37 €	619.343,75 €
	397,44 €/p	408,63 €/p	415,82 €/p	415,94 €/p	402,60 €/p	302,12 €/p
Redistribution (Forward Redistribution)	474,83 m^2	466,41 m^2	464,56 m^2	499,08 m^2	454,04 m^2	887,29 m^2
	835 p	795 p	770 p	815 p	795 p	2.050 p
	37,57 %	42,59 %	42,29 %	43,99 %	41,07 %	16,03 %
	0,66 p/m^2	0,73 p/m^2	0,70 p/m^2	0,72 p/m^2	0,72 p/m^2	0,47 p/m^2
	331.864,38 €	324.856,88 €	320.178,75 €	338.991,88 €	320.069,37 €	619.343,75 €
	397,44 €/p	408,63 €/p	415,82 €/p	415,94 €/p	402,60 €/p	302,12 €/p

Figure 27. *Results of the experiment with a low volume of stock and a high rotation of stock.*

Although the results of this experiment are not so great in terms of improvement of the solution in comparison with a single-deep storing system, indicator 1,2,3 and 5 reduce the space utilisation used by the single-deep storing system.

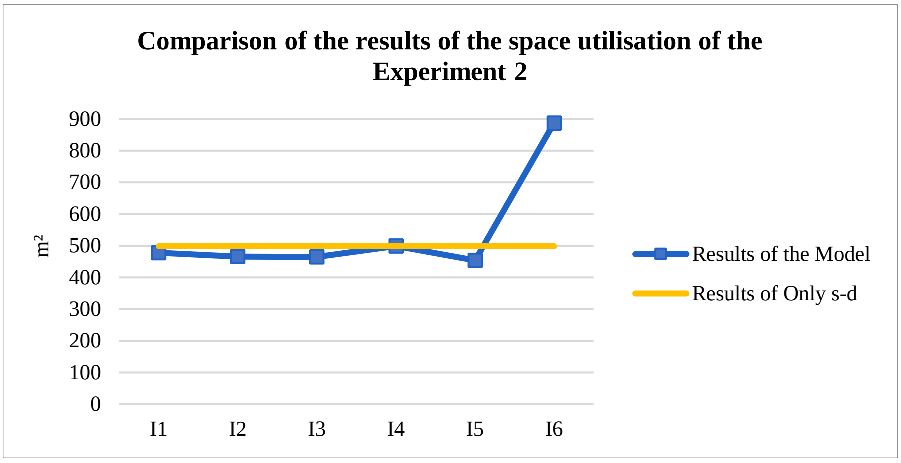


*Figure 28. Comparison of the **space utilisation** of the experiment with a low volume of stock and a high rotation.*

The following table shows the comparison between the output results of the Model in relation with a storage system of only single-deep lanes.

Comparison of the best results of the Experiment 2		
Results I5	Only single deep	Improvement I5
454,04 m²	498,71 m²	8,96 %
795 p	650 p	22,31 %
41,07 %	47,31 %	13,19 %
0,72 p/m²	0,62 p/m²	16,13 %
320.069,37 €	298.256,25 €	7,31 %
402,60 €/p	458,86 €/p	12,26 %

Table 31. Comparison of the best results of the experiment with a low volume of stock and a high rotation of stock with an only single deep storage system.

It is important to highlight that in low volume of stock and high rotation of stock circumstances is when the Optimization Model performs worst. This experiment is the most challenging one because if the stock levels and incoming levels are low and the rotation is high, the multi-deep lanes are less space efficiency. However, the Indicator 5 is clearly the one that performs greater results and performs a reduction of the space of almost 10%.

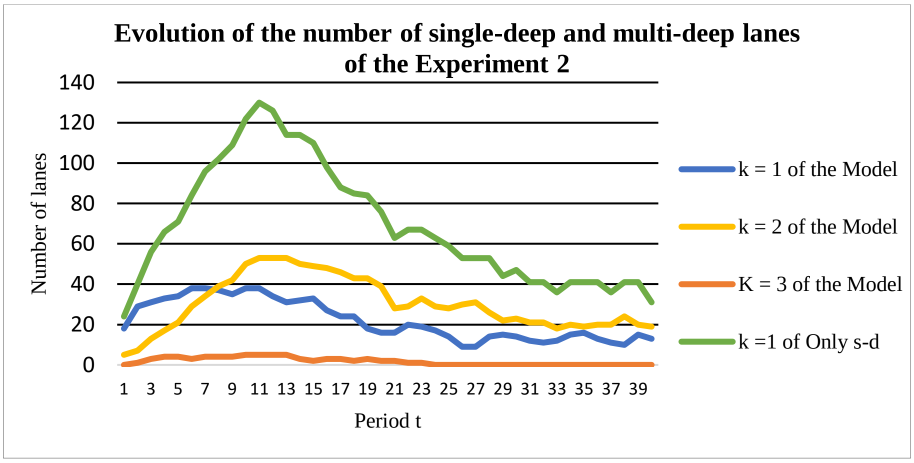


Figure 29. *Comparison of the* ***evolution of the number*** *of single-deep and multi-deep lanes of the experiment with a low volume of stock and a high rotation.*

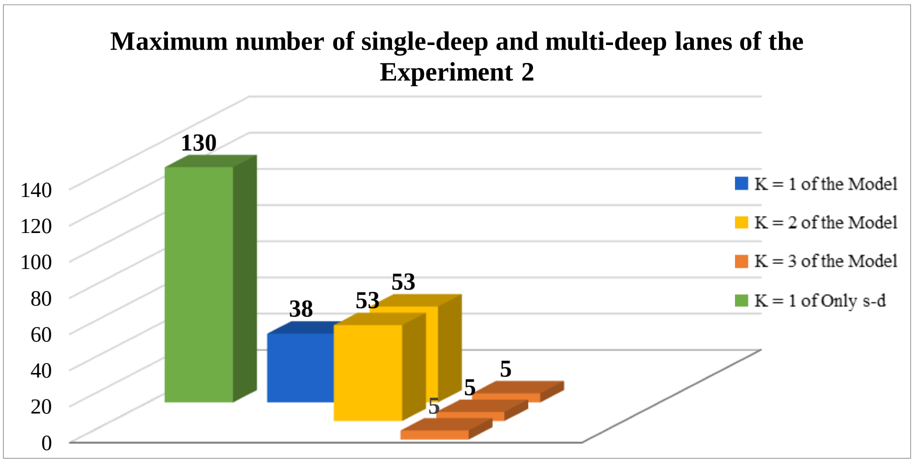


Figure 30. *Comparison of the* ***maximum number*** *of single-deep and multi-deep lanes of the experiment with a low volume of stock and a high rotation.*

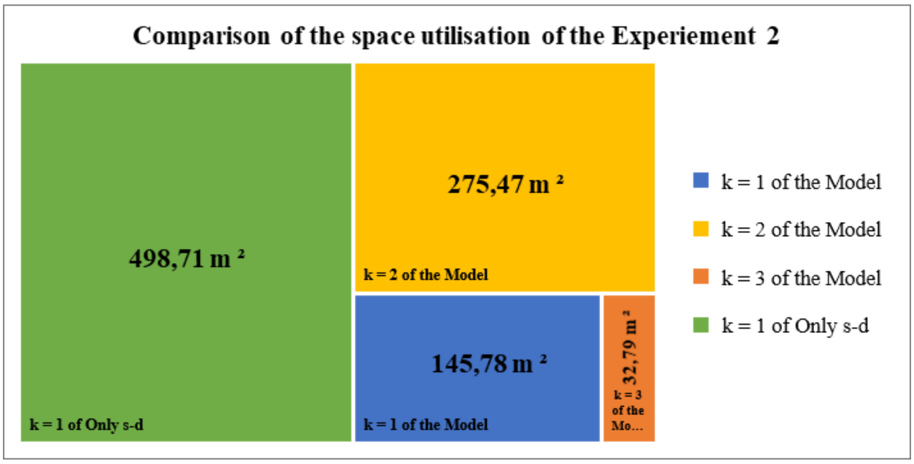


Figure 31. *Comparison of the* ***space utilisation*** *of the experiment with a low volume of stock and a high rotation of stock.*

5.3.3. Experiment with a high volume of stock and a low rotation of stock

	Results of the Experiment 3					
	I1	I2	I3	I4	I5	I6
Local Optimization	1.056,58 m²	1.151,62 m²	811,8 m²	1097,78 m²	1.105,83 m²	1.464,33 m²
	2.790 p	1.710 p	1.900 p	2.505 p	2.880 p	3.415 p
	33,17%	57,56 %	48,71 %	36,95 %	32,14 %	27,28 %
	0,88 p/m²	1,36 p/m²	1,14 p/m²	0,84 p/m²	0,84 p/m²	0,64 p/m²
	767.788,75 €	518.611,25 €	579.500,00 €	754.243,12 €	796.715,00 €	982.216,87 €
	275,19 €/p	303,28 €/p	305,00 €/p	301,10 €/p	276,64 €/p	287,62 €/p
Relocation	758,83 m²	1.015,24 m²	807,10m²	882,83 m²	897,93 m²	1.710,47 m²
	1.860 p	2.520 p	1.810 p	1.870 p	2.280 p	3.810 p
	49,76 %	36,73 %	51,13%	49,49 %	40,59 %	24,29 %
	1,22 p/m²	0,91 p/m²	1,15 p/m²	1,05 p/m²	1,03 p/m²	0,54 p/m²
	551.817,50 €	518.611,25 €	570.448,75 €	602.916,25 €	648.965,00 €	1.119.536,25€
	296,68 € /p	303,28 €/p	315,17 €/p	322,42 €/p	284,63 €/p	293,84 €/p
Redistribution (Initial Peak Redistribution)	692,88 m²	724,06 m²	707,60 m²	762,05 m²	708,22 m²	1.548,98 m²
	1645 p	1.710 p	1590 p	1.590 p	1.765 p	3.535 p
	57,59 %	57,56 %	59,67 %	59,64 %	54,95 %	27,07 %
	1,37 p/m²	1,36 p/m²	1,34 p/m²	1,24 p/m²	1,37 p/m²	0,62 p/m²
	503.188,12 €	518.611,25 €	505.301,25 €	522.926,25 €	517.260,62 €	1.026.539,37 €
	305,89 €/p	303,28 €	317,80 €/p	328,88 €/p	293,07 €/p	290,39 €/p
Redistribution (Backward Redistribution)	688,79 m²	724, 06 m²	694,49 m²	746,71 m²	708,22 m²	1.540,81m²
	1.630 p	1.705	1.560 p	1.570 p	1.765 p	3.505 p
	57,89%	57,66%	60,70 %	60,32 %	54,60 %	26,41 %
	1,37 p/m²	1,36 p/m²	1,36p/m²	1,27 p/m²	1,36 p/m²	0,60 p/m²
	500.296,25 €	517.780,63 €	497.842,5 €	514.453,75 €	517.260,62 €	1.020.555,63 €
	306,93 €/p	303,68 €/p	319,13 €/p	327,68 €/p	293,07 €/p	291,17 €/p
Redistribution (Forward Redistribution)	688,79 m²	724, 06 m²	694,49 m²	746,71 m²	708,22 m²	1.540,81m²
	1.630 p	1.705 p	1.560 p	1.570 p	1.765 p	3.505 p
	57,89%	57,66%	59,33 %	58,95 %	54,60 %	26,41 %
	1,37 p/m²	1,36 p/m²	1,33 p/m²	1,24 p/m²	1,36 p/m²	0,60 p/m²
	500.296,25 €	517.780,63 €	497.842,5 €	514.453,75 €	517.260,62 €	1.020.555,63 €
	306,93 €/p	303,68 €/p	319,13 €/p	327,68 €/p	293,07 €/p	291,17 €/p

Table 32. *Results of the experiment with a high volume of stock and a low rotation of stock.*

In this case, the indicators that provides better results are the Indicator 1 and the Indictor 3. The results of the Indicator 6 are far away of a reasonable solution.

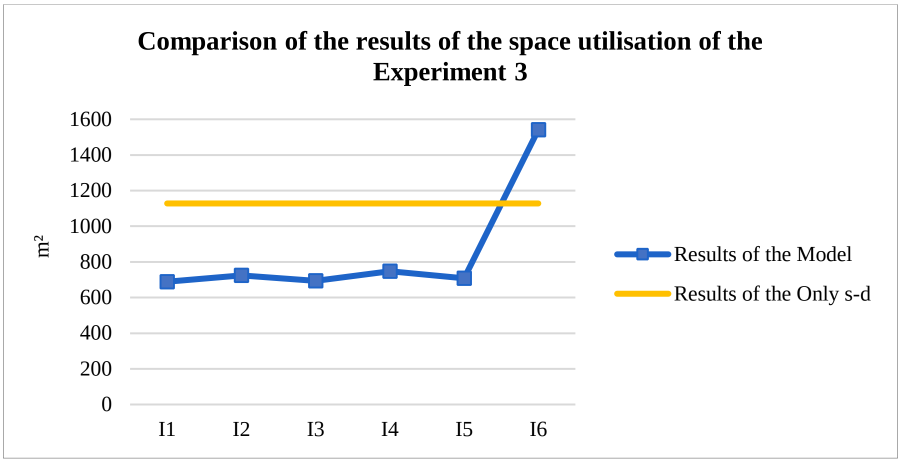


Figure 32. Comparison of the **space utilisation** of the experiment with a high volume of stock and a low rotation.

Comparison of the best results of the Experiment 3		
Results I1	**Only single deep**	**Improvement I1 vs s-d**
688,79 m^2	1.127,86 m^2	38,93%
1.630 p	1.470 p	10,88%
57,89%	62,96 %	-8,05%
1,37 p/m^2	0,82 p/m^2	67,07%
500.296,25 €	617.748,75 €	19,01%
306,93 €/p	420,24 €/p	26,96%

Table 33. Comparison of the best results of the experiment with a high volume of stock and a low rotation of stock of the Model vs Only single-deep storage system.

The results of the experiment 3 are the most satisfactory, with reductions of the space utilization of almost 40% and cost reductions beyond 25%.

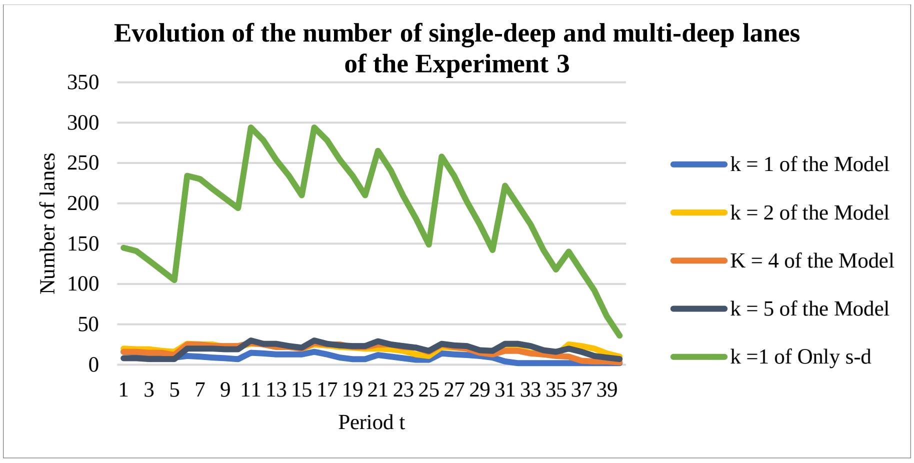


Figure 33. Comparison of the **evolution of the number** of single-deep and multi-deep lanes of the experiment with a high volume of stock and a low rotation.

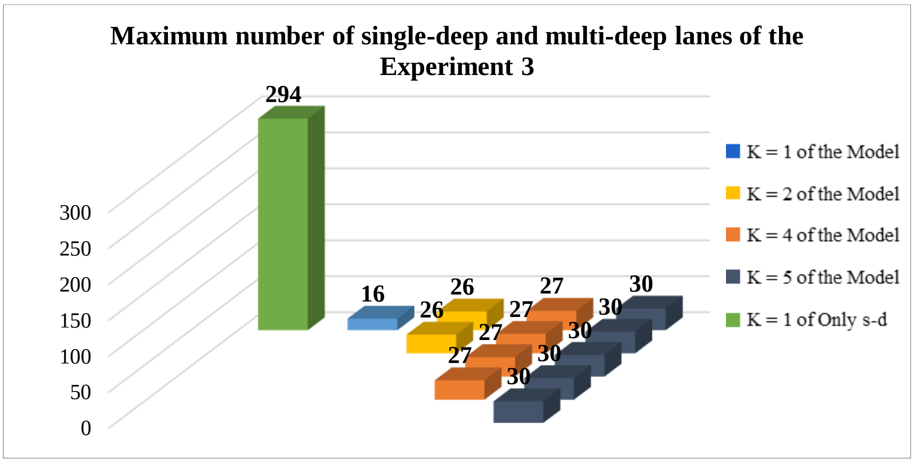


Figure 34. *Comparison of the* ***maximum number*** *of single-deep and multi-deep lanes of the experiment with a high volume of stock and a low rotation.*

As it can be seen in *Figure 34*, it starts to be necessary the usage of lane depths greater than 3. This feature wasn't observable in Experiment 1 or 2 because their levels of incoming and stock where low. Conversely, here there are more lanes of depth $k = 4$ and $k = 5$ than single-deep or multi-deep of $k = 2$ that is one of the most used lane depths. Also, looking at *Figure 34* is easy to realize the broad difference between using only single-deep lanes instead of using a combination of single-deep and multi-deep lanes.

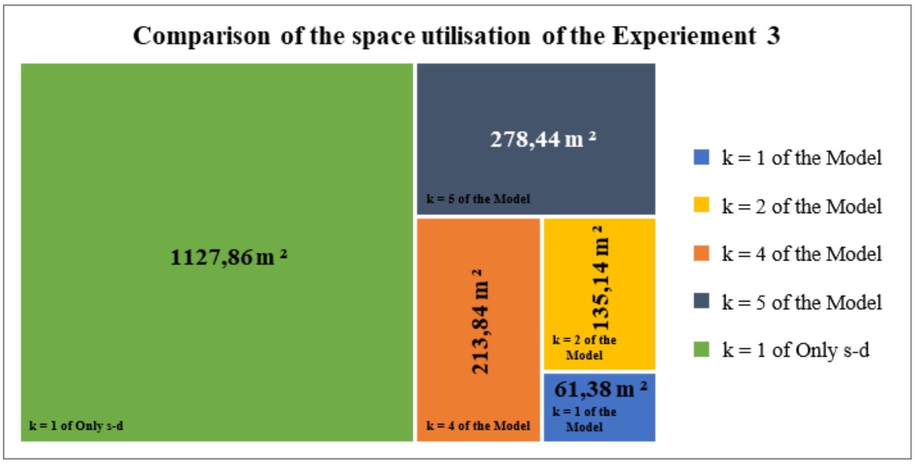


Figure 35. *Comparison of the* ***space utilisation*** *of the experiment with a high volume of stock and a low rotation of stock.*

5.3.4. Simulations with a high volume of stock and a high rotation of stock

	Results of the experiment 4					
	I1	I2	I3	I4	I5	I6
Local Optimization	1984,46 m²	2.153,74 m²	2.192,60 m²	2.191,37 m²	1.965,40 m²	2.109,81 m²
	5080 p	5.520 p	5.490 p	5.440 p	5.010 p	5.595 p
	34,07%	31,36 %	31,53 %	31,82 %	34,55 %	34,84 %
	0,87 p/m²	0,80 p/m²	0,79 p/m²	0,79 p/m²	0,88 p/m²	0,92 p/m²
	1.379.627,5 €	1.494.672,50 €	1.510.201,25 €	1.504.882,50 €	1.364.998,75 €	1.469.756,88 €
	271,58 €	270,77 €/p	275,08 €/p	276,63 €/p	272,45 €/p	262,69 €/p
Relocation	1803,03 m²	1.804,65 m²	1.769,01 m²	1938,92 m²	1.803,28 m²	2.445,05 m²
	3.950 p	3.865 p	3.525 p	3.740 p	3.860 p	5.590 p
	44,44%	46,50 %	52,12 %	49,42 %	46,38 %	32,56 %
	0,97 p/m²	1,00 p/m²	1,04 p/m²	0,95 p/m²	0,99 p/m²	0,74 p/m²
	1.209.218,75 €	1.199.073,13 €	1.148.053,13 €	1.239.857,5 €	1.192.042,5 €	1.605.826,25 €
	306,13 €/p	310,24 €/p	325,69 €/p	331,51 €/p	308,82 €/p	287,27 €/p
Redistribution (Initial Peak Redistribution)	1.430,92 m²	1.468,67 m²	1.549,35 m²	1.569,8m²	1.534, 25 m²	2.429,46 m²
	3.040	3.140 p	3.000 p	3.030 p	3.190 p	5.560 p
	65,3%	60,61 %	69,56 %	67,60 %	61,02 %	32,78 %
	1,40 p/m²	1,30 p/m²	1.35 p/m²	1,30 p/m²	1,27 p/m²	0,75 p/m²
	954.145 €	986.332,5 €	1.009.275 €	1.019.446,25 €	1.012.226,25 €	1.595.930,00 €
	313,86 €	314,12 €/p	336,425 €/p	336,45 €/p	317,31 €/p	287,04 €/p
Redistribution (Backward Redistribution)	1416,69 m²	1.390,70 m²	1.507,77 m²	1.370,90 m²	1.495,76 m²	2.377,98 m²
	3.040 p	2.990 p	2.920 p	2.690 p	3.125 p	5.480 p
	65,30 %	62,94 %	69,72 %	69,01 %	62,82 %	34,09 %
	1,40 p/m²	1,35 p/m²	1,35 p/m²	1,35 p/m²	1,31 p/m²	0,79 p/m²
	954.145,00 €	939.251,25 €	986.485,00 €	905.751,25 €	990.328,13 €	1.566.190,00 €
	313,86 €/p	314,13 €/p	337,84 €/p	336,71 €/p	316,91 €/p	€285,80 /p
Redistribution (Forward Redistribution)	1416,69 m²	1.390,70 m²	1.507,77 m²	1.370,90 m²	1.495,76 m²	2.377,98 m²
	3.040 p	2.990 p	2.920 p	2.690 p	3.125 p	5.480 p
	65,30 %	62,94 %	69,72 %	69,01 %	62,82 %	34,09 %
	1,40 p/m²	1,35 p/m²	1,35 p/m²	1,35 p/m²	1,31 p/m²	0,79 p/m²
	954.145,00 €	939.251,25 €	986.485,00 €	905.751,25 €	990.328,13 €	1.566.190,00 €
	313,86 €/p	314,13 €/p	337,84 €/p	336,71 €/p	316,91 €/p	€285,80 /p

Table 34. Results of the experiment with a high volume of stock and a high rotation of stock.

Comparison of the best results of the Experiment 4		
Results I4	**Only single deep**	**Improvement**
1.370,90 m^2	1.898,94 m^2	27,81 %
2.690 p	2.475 p	8,69 %
69,01 %	69,90 %	-1,27%
1,35 p/m^2	0,91 p/m^2	48,35 %
905.751,25 €	1.000.321,88 €	9,45 %
336,71 €/p	407,81 €/p	17,43 %

Table 35. *Comparison of the best results of the experiment with a high volume of stock and a high rotation of stock with an only single-deep storage system*

The performance of the Model in Experiment4 is also considerably advantageous compared with single-deep storage systems. It is remarkable the difference in the surface efficiency, where the Model makes an improvement of practically the 50%.

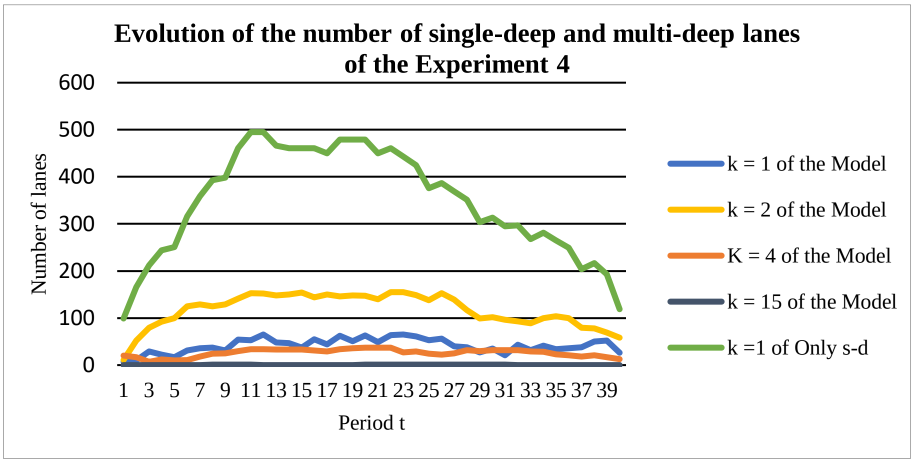


Figure 36. *Comparison of the* ***evolution of the number*** *of single-deep and multi-deep lanes of the experiment with a high volume of stock and a high rotation.*

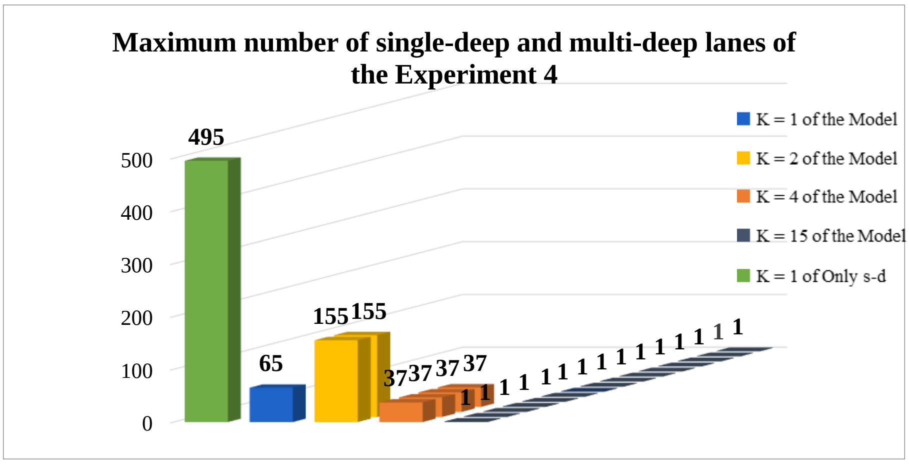


Figure 37. *Comparison of the* ***maximum number*** *of single-deep and multi-deep lanes of the experiment with a high volume of stock and a high rotation.*

The outcome of Experiment 4 shows for the first time a lane depth of 15 rows. Even though at the end of the Local Optimization process there were 27 lanes of *k = 15*, when the program balances the distribution to get a better space utilisation the products located in these lanes are relocated in lanes ok k=2 and k=3.

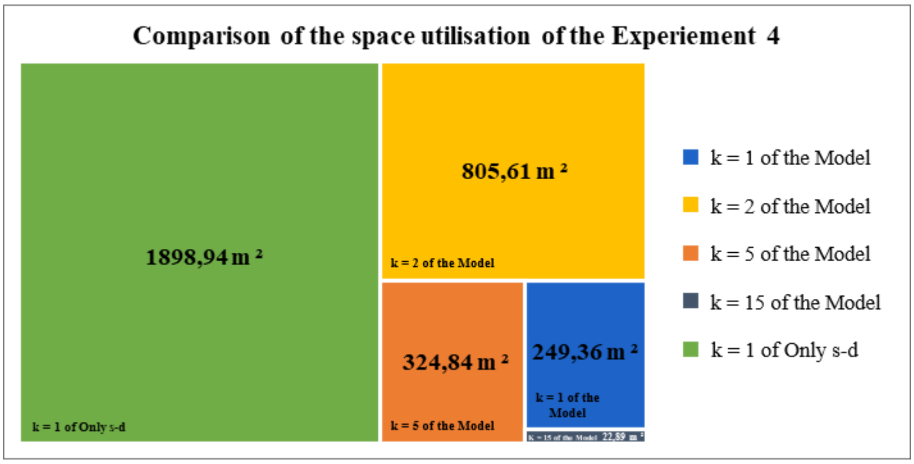


Figure 38. *Comparison of the* ***space utilisation*** *of the experiment with a high volume of stock and a high rotation of stock.*

6. Conclusions

The Optimization Model for multi-deep storage systems is evaluated through four different scenarios with different kind of stock and rotation. We obtain the following insights:

1. *The results of the Model are better than an Only single-deep storage system*

Primarily, the final results of the four simulations carried out are better in terms of space utilisation than a storage system consisting of only single-deep lanes. With these results, can be confirmed that the model gives a properly result. The improvements in terms of space utilisation are at least 8,96% (Experiment 2: A low volume of stock and a high rotation) compared with single-deep storing system. Total space utilisation reductions are even better with other simulations, where the Model achieves reductions of 38,93% (Experiment 3: A high volume of stock and a low rotation). As it was expected and based on the results, it can be deducted that the Model performs better with low rotation of stock than for high rotation of stock and for high volumes of stock than for ow volumes of stock. However, it does not mean that the Model will always give better results as more simulations are required to confirm this result.

With high stock levels that require a large space to store all the inventory, the investment costs of the storage systems can be considerably high. The right combination of single-deep and multi-deep lanes of different depth can contribute to enormous cost savings. For instance, grounded on the results of the Experiment 4, an investment that can soar till approximately 1.000.000 €, choosing the combination of the Model the saving can be of more or less 100.000 €.

2. *Forward redistribution process low yield*

After the realisation of the experiments, there have not been results indicating a progress in the implementation of the "Forward Redistribution Process". The design of this process was based on very specific circumstances where there were two peaks of usage of lanes of the same depth k and the second peak was a stumbling block for the improvement of the redistribution process.

To go on with simulations with a wide range of different products we recommend to not carry out this process. The main reason is the consumption of time that need the algorithm to perform this process is very high and there if there is a large number of different SKU carry out this process would mean probably a waste of resources.

3. *Indicators performance*

In all the experiments the indicator 6 has outperformed the results of the other indicators. This indicator is not suitable for the model created in this Master Thesis. It is important to highlight that it does not mean that the cube utilisation is a bad formula when the lane depth of a storage system is decided upon, it means that it does not fit with the process followed by this model but could work properly through the application of another approach.

Based on the outcome of the experiments, indicator 1,2,3,4 and 5 deliver similar results. Trying to find a correlation between the indicator's performances and the stock characteristics, it is likely that the indicator 1 performs better in circumstances of low rotation of stock and that the indicator 5 performs more favourable in scenarios with low stock.

4. *Improvement of the current model*

Following the same approach of the Model, there are several modifications that could be simulated to enhance the performance of the model. Below there

- Setting product classes at the beginning in function of the incoming number of pallets and calculating the indicator for all of them at the same time.
- The possible combinations to calculate the indicator are limitless. The assumptions make in the model to calculate the indicator take into consideration the products that are still stored in the warehouse in a local level. Another research line could be to increase the scope of the calculation of the indicator of product i and taking into account the lanes used since day $t = 1$ for all the products.

5. *Further investigation*

Further research development is expected on the other multi-deep storing system like shuttle racking that allow to store different SKU i to the same multi-deep lane. This way the space utilisation, the storage efficiency and the surface efficiency of the storing system could increase significantly compared with the results obtained with the drive-in storing systems.

List of Figures

List of Tables

Bibliographical references

Accorsi, R., Baruffaldi, G. and Manzini, R., 2017, *"Design and manage deep lane storage system layout. An iterative decision-support model"*, Int J Adv. Manuf. Technol. 92:57–67.

Baker, P., Canessa M., 2009, *"Warehouse design: A structured approach"*, European Journal of Operational Research, 193(2), 425-436.

Bartholdi, J. J., III and Hackman, S. T., 2017, *"Warehousing & Distribution Science"*, Release 0.98, www.warehouse-science.com.

Cormier, Gilles, and Eldon A. Gunn, 1992, *"A review of warehouse models"*, European Journal of Operational Research 58(1), 3-13.

Ferrara, A., Gebennini, E., Grassi, A., and Rimini, B., 2011, *"Efficiency estimation of LIFO storage systems managed with Automated Guided Vehicles (AGVs)"*, Proc. of the International Conference on Production Research, July 31-August 4, Stuttgart, Germany.

Frazelle, E., 2002a. "*Supply Chain Strategy: The Logistics of Supply Chain Management"*, McGraw-Hill, New York.

Frazelle, E., 2002b. *"World-class Warehousing and Material Handling"*. McGraw-Hill, New York.

Gu, J.X., Goetschalckx Gu, J., Goetschalckx, M., and McGinnis, L. F., 2010, *"Research on warehouse design and performance evaluation: A comprehensive review"*, European Journal of Operational Research, 177(1), 1-21.

Gu, J., Goetschalckx, M., and McGinnis, L. F., 2010, *"Research on warehouse design and performance evaluation: A comprehensive review"*, European Journal of Operational Research, 203(3), 539-549.

Goetschalckx,M., and Ratliff, H.D., 1991, *"Optimal lane depths for single and multiple products in block stacking storage systems,"* IIE Transactions, 23(3), 245-258.

Kay GM, 2009, *"Lecture Notes for Production System Design"*. Department of Industrial and Systems Engineering, North Carolina State University.

Kay, M. G. (2015). *"Warehousing"* Fitts Dept. of Industrial and Systems Engineering North Carolina State University.

Kind DA, 1965, *"Measuring warehouse space utilization"*, Transportation and Distribution Management 7(5):23–33.

Kind DA, 1975, *"Elements of space utilization"*, Transportation and Distribution Management 15(1):29–34.

Kim KH, Hong G-P, 2006, *"A heuristic rule for relocating blocks"*, Comput Oper Res 33(4):940–954.

Kooy ED (1975) *"Making better use of available warehouse space"*, Ind Eng 13(10):26–30.

Matson JO, White JA, 1981, *"Storage system optimization"*, Production and distribution research Centre report, 81–89. Georgia Institute of Technology, Atlanta, Georgia.

Matson JO, Sonnentag JJ, White JA, Imhoff RC, 2014, *"An analysis of block stacking with lot splitting"*. IIE Annual Conference and Expo 2014, 497–506.

Nima Zaerpour, Yugang Yu & René B.M. de Koster (2017) "*Optimal two-class-based storage in a live-cube compact storage system"*, IISE Transactions, 49:7, 653-668.

B. Rouwenhorst, B. Reuter, V. Stockrahm, G.J. van Houtum, R.J. Mantel, W.H.M. Zijm, 1999, "*Warehouse design and control: Framework and literature review"*, European Journal of Operational Research,Volume 122, Issue 3,2000, Pages 515-533, ISSN 0377-2217.

Sonnentag JJ, Imhoff RC, White JA, Matson JO, 2014, *"A consideration of the block stacking multi-product storage problem"*. IIE Annual Conference and Expo 2014, 3221–3230.

White JA, Sonnentag JJ, Matson JO, 2013, *"New insights regarding block stacking"*. IIE Annual Conference and Expo 2013, 1225–1234.

Annex A

A.1. Incoming Distribution

A.1.1. Priority single-deep

A.1.1.1. "Prior_sd_tot1"

$If\ (l_{sd} \cdot z) - s_{itsd} > q_{it}$:

$$s_{q_{it}sd} = q_{it}$$

$$l_{q_{it}sd} = \frac{s_{q_{it}sd}}{z}$$

$$s_{itsd} = s_{itsd} + q_{it}$$

$$l_{itsd} = \frac{s_{itsd}}{z}$$

$$Indicator\_prior\_sd\_tot1 = Calculate\ indicator$$

A.1.1.2. "Prior_sd_tot_2

$If\ 0 < (l_{sd} \cdot z) - s_{itsd} < q_{it}$:

$$s_{q_{it}sd} = l_{sd} \cdot z - s_{itsd}$$

$$l_{q_{it}sd} = \frac{s_{q_{it}sd}}{z}$$

$$s_{q_{it}mdk} = q_{it} - (l_{sd} \cdot z - s_{itsd})$$

$$l_{q_{it}mdk} = \frac{s_{q_{it}mdk}}{k \cdot z}$$

$$k_{q_{it}mdk} = k$$

$$real = \frac{s_{q_{it}mdk}}{k \cdot z}$$

$$integer = \frac{s_{q_{it}mdk}}{k \cdot z}$$

$$b_{q_{it}mdk} = (real - integer) \cdot k \cdot z$$

$$c_{q_{it}mdk} = \frac{b_{q_{it}mdk}}{z}$$

$$s_{itsd} = s_{itsd} + s_{q_{it}sd}$$

$$l_{itsd} = \frac{s_{itsd}}{z}$$

$$s_{itmd} = s_{itmd} + s_{q_{it}mdk}$$

$$l_{itmd} = l_{itmd} + l_{q_{it}mdk}$$

$$Indicator\_prior\_sd\_tot2 = Calculate\ indicator$$

A.1.2. Priority multi-deep

A.1.2.1. “Prior_md_tot1”

$s_{q_{it}mdk} = q_{it}$

$l_{q_{it}mdk} = \frac{s_{q_{it}mdk}}{k \cdot z}$

$k_{q_{it}mdk} = k$

$real = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$integer = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$b_{q_{it}mdk} = (real - integer) \cdot k_{q_{it}mdk} \cdot z$

$c_{q_{it}mdk} = \frac{b_{q_{it}mdk}}{z}$

$s_{itmd} = s_{itmd} + s_{q_{it}mdk}$

$l_{itmd} = l_{itmd} + l_{q_{it}mdk}$

$Indicator\_prior\_md\_tot\_1 = Calculate\ indicator$

A.1.2.2. “Prior_md_tot2”

$if\ 0 < b_{q_{it}mdk} \leq (l_{sd} \cdot z) - s_{itsd}:$

$s_{q_{it}sd} = b_{q_{it}mdk}$

$l_{q_{it}sd} = \frac{s_{q_{it}sd}}{z}$

$s_{q_{it}mdk} = s_{q_{it}mdk} - b_{q_{it}mdk}$

$l_{q_{it}mdk} = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$s_{itsd} = s_{itsd} + s_{q_{it}sd}$

$l_{itsd} = \frac{s_{itsd}}{z}$

$s_{itmd} = s_{itmd} - b_{q_{it}mdk}$

$l_{itmd} = l_{itmd} - 1$

$real = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$integer = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$b_{q_{it}mdk} = (real - integer) \cdot k_{q_{it}mdk} \cdot z$

$c_{q_{it}mdk} = \frac{b_{q_{it}mdk}}{z}$

$Indicator\_prior\_md\_tot\_2 = Calculate\ indicator$

A.1.2.3. "Prior_md_tot3"

$if\ b_{q_{it}mdk} \geq (l_{sd} \cdot z) - s_{itsd} > 0$:

$s_{q_{it}sd} = l_{sd} \cdot z - s_{itsd}$

$l_{q_{it}sd} = \frac{s_{q_{it}sd}}{z}$

$s_{q_{it}mdk} = s_{q_{it}mdk} - \left((l_{sd} \cdot z) - s_{itsd}\right)$

$lanes\ a = l_{q_{it}md}$

$l_{q_{it}mdk} = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$lanes\ b = l_{q_{it}md}$

$lanes\ c = lanes\ a - lanes\ c$

$real = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$integer = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$b_{q_{it}mdk} = (real - integer) \cdot k_{q_{it}mdk} \cdot z$

$c_{q_{it}mdk} = \frac{b_{q_{it}mdk}}{z}$

$s_{itsd} = s_{itsd} + s_{q_{it}sd}$

$l_{itsd} = \frac{s_{itsd}}{z}$

$s_{itmd} = s_{itmd} - \left((l_{sd} \cdot z) - s_{itsd}\right)$

$l_{itmd} = l_{itmd} - lanes\ c$

$Indicator\_prior\_md\_tot\_3 = Calculate\ indicator$

A.2 Inventory Control

$cont = 0$

$d = d_{it}$

$while\ cont > d_{it}$

A.2.1. Stock in single-deep and multi-deep lanes

$if\ s_{q_{it}sd} > 0\ and\ s_{q_{it}mdk} > 0:$

FUNC PRIOR SD

$if\ 0 < d \leq s_{q_{it}sd}:$

$cont = cont + d$

$s_{q_{it}sd} = s_{q_{it}sd} - d$

$l_{q_{it}sd} = \frac{s_{q_{it}sd}}{z}$

$s_{itsd} = s_{itsd} - d$

$l_{itsd} = \frac{s_{itsd}}{z}$

$Indicator = Calculate\ indicator$

$Indicator\_prior\_sd = Indicator$

$else$:

$cont = cont + s_{q_{it}mdk}$

$d = d - s_{q_{it}mdk}$

$s_{itsd} = s_{itsd} - s_{itsd}$

$l_{itsd} = \frac{s_{itsd}}{z}$

$s_{q_{it}sd} = 0$

$l_{q_{it}sd} = 0$

$Indicator = Calculate\ indicator$

$Indicator\_prior\_sd = Indicator$

$if\ 0 < d \leq s_{q_{it}mdk}$:

$cont = cont + d$

$s_{itmd} = s_{itmd} - d$

$s_{q_{it}mdk} = s_{q_{it}mdk} - d$

$lanes\ a = l_{q_{it}md}$

$l_{q_{it}mdk} = \frac{s_{q_{it}mdk}}{k \cdot z}$

$lanes\ b = l_{q_{it}md}$

$lanes\ c = lanes\ a - lanes\ c$

$l_{itmd} = l_{itmd} - lanes\ c$

$real = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$integer = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$b_{q_{it}mdk} = (real - integer) \cdot k_{q_{it}mdk} \cdot z$

$c_{q_{it}mdk} = \frac{b_{q_{it}mdk}}{z}$

$Indicator = Calculate\ indicator$

$Indicator\_prior\_sd = Indicator$

$else$:

$cont = cont + s_{q_{it}mdk}$

$d = d - s_{q_{it}mdk}$

$s_{itmd} = s_{itmd} - s_{q_{it}mdk}$

$l_{itmd} = l_{itmd} - l_{q_{it}mdk}$

$Indicator = Calculate\ indicator$

$Indicator\_prior\_sd = Indicator$

FUNC PRIOR MD

$if\ 0 < d \leq s_{q_{it}mdk}$:

$cont = cont + d$

$s_{q_{it}mdk} = s_{q_{it}mdk} - d$

$lanes\ a = l_{q_{it}md}$

$l_{q_{it}mdk} = \frac{s_{q_{it}mdk}}{k \cdot z}$

$lanes\ b = l_{q_{it}md}$

$lanes\ c = lanes\ a - lanes\ c$

$s_{itmd} = s_{itmd} - d$

$l_{itmd} = l_{itmd} - lanes\ c$

$real = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$integer = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$b_{q_{it}mdk} = (real - integer) \cdot k_{q_{it}mdk} \cdot z$

$c_{q_{it}mdk} = \frac{b_{q_{it}mdk}}{z}$

$Indicator = Calculate\ Indicator$

$Indicator\_prior\_md = Indicator$

$else$:

$cont = cont + d$

$d = d - s_{q_{it}mdk}$

$s_{itmd} = s_{itmd} - s_{q_{it}mdk}$

$l_{itmd} = l_{q_{it}mdk} - l_{q_{it}mdk}$

$s_{q_{it}mdk} = 0$

$l_{q_{it}mdk} = 0$

$if\ 0 < d \leq s_{q_{it}sd}$:

$cont = cont + d$

$s_{itsd} = s_{itsd} - d$

$s_{q_{it}sd} = s_{q_{it}sd} - d$

$l_{itsd} = \frac{s_{itsd}}{z}$

$real = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$

$$integer = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$$

$$b_{q_{it}mdk} = (real - integer) \cdot k_{q_{it}mdk} \cdot z$$

$$c_{q_{it}mdk} = \frac{b_{q_{it}mdk}}{z}$$

$else$:

$$cont = cont + s_{q_{it}sd}$$

$$d = d - s_{q_{it}sd}$$

$$s_{itsd} = s_{itsd} - s_{q_{it}sd}$$

$$l_{itsd} = \frac{s_{itsd}}{z}$$

$$indicator = Caclulate\ indicator$$

$$indicator\_prior\_md = indicator$$

$if\ indicator\_prior\_sd \leq Indciator\_prior\_md$:

FUNC PRIOR SD

$else$:

FUNC PRIOR MD

A.2.2. Stock in only single-deep or only multi-deep lanes

$else$:

$if\ s_{q_{it}sd} > 0$:

FUNC SD

$if\ 0 < d \leq s_{q_{it}sd}$:

$$cont = cont + d$$

$$s_{q_{it}sd} = s_{q_{it}sd} - d$$

$$l_{q_{it}sd} = \frac{s_{q_{it}sd}}{z}$$

$$s_{itsd} = s_{itsd} - d$$

$$l_{itsd} = \frac{s_{itsd}}{z}$$

$else$:

$$cont = cont + s_{q_{it}sd}$$

$$d = d - s_{q_{it}sd}$$

$$s_{itsd} = s_{itsd} - d$$

$$l_{itsd} = \frac{s_{itsd}}{z}$$

$else$:

FUNC MD

$if\ 0 < d \leq s_{q_{it}mdk}$

$$cont = cont + d$$

$$s_{q_{it}mdk} = s_{q_{it}mdk} - d$$

$$lanes\ a = l_{q_{it}md}$$

$$l_{q_{it}mdk} = \frac{s_{q_{it}mdk}}{k \cdot z}$$

$$lanes\ b = l_{q_{it}md}$$

$$lanes\ c = lanes\ a - lanes\ c$$

$$s_{itmd} = s_{itmd} - d$$

$$l_{itmd} = l_{itmd} - lanes\ c$$

$$real = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$$

$$integer = \frac{s_{q_{it}mdk}}{k_{q_{it}mdk} \cdot z}$$

$$b_{q_{it}mdk} = (real - integer) \cdot k_{q_{it}mdk} \cdot z$$

$$c_{q_{it}mdk} = \frac{b_{q_{it}mdk}}{z}$$

$else$:

$$cont = cont + s_{q_{it}mdk}$$

$$d = d - s_{q_{it}mdk}$$

$$s_{itmd} = s_{itmd} - s_{q_{it}mdk}$$

$$l_{itmd} = l_{itmd} - l_{q_{it}mdk}$$

Annex B

B.1.Incoming and demand of the Experiments

B.1.1. Incoming and demand of the Experiment 1

	q_{it}																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	5	5	5	13	5	7	7	10	5	5	5	5	2	3	3	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	15	10	10	10	0	8	0	0	0	5	10	10	0	3	0	0	0	1	2	2
7	0	0	0	0	0	0	15	15	0	0	0	0	0	0	5	5	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	4	3	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	30	30	0	0	0	0	0	0	10	10	0	0	0	0	0	0	2	2
17	0	0	0	0	0	0	15	15	0	0	0	0	0	0	5	5	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	10	20	20	0	0	0	0	0	5	10	10	0	3	0	0	0	1	3	3
27	0	0	0	0	0	8	15	15	0	0	0	0	0	0	5	5	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	4	4	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	10	10	0	0	0	0	0	0	5	5	0	0	0	0	0	0	1	1
37	0	0	0	0	0	0	6	6	0	0	0	0	0	0	2	2	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

d_{it}																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	1	1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
12	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
13	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
14	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
15	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	1	1	1	1
16	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
17	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
18	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
19	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
20	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
21	3	3	3	3	4	4	4	4	2	2	2	2	2	2	2	2	0	0	0	0
22	3	3	3	3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0
23	3	3	3	3	4	4	4	4	2	2	2	2	2	2	2	2	0	0	0	0
24	3	3	3	3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0
25	3	3	3	3	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2
26	3	3	3	3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0
27	3	3	3	3	4	4	4	4	2	2	2	2	2	2	2	2	0	0	0	0
28	3	3	3	3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0
29	3	3	3	3	4	4	4	4	2	2	2	2	2	2	2	2	0	0	0	0
30	3	3	3	3	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2
31	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
32	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
33	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
34	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
35	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	1	1	1	1
36	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
37	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
38	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
39	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
40	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Table 36. *Demand of the Experiment 1*

B.1.2. Incoming and demand of the experiment 2

	q_{it}																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	10	15	7	7	7	10	5	5	5	5	2	3	3	2	1	1	1
2	0	0	10	20	0	8	15	15	0	5	10	10	0	3	5	5	0	1	2	2
3	30	30	10	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2	0	0
4	0	0	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2
5	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
6	0	10	20	20	0	8	15	15	0	5	10	10	0	3	5	5	0	1	2	2
7	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2	0	0
8	0	0	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2
9	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
10	0	10	20	20	0	8	15	15	0	5	10	10	0	3	5	5	0	1	2	2
11	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2	0	0
12	0	0	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2
13	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
14	0	10	20	20	0	8	15	15	0	5	10	10	0	3	5	5	0	1	2	2
15	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2	0	0
16	0	0	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2
17	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
18	0	10	20	20	0	8	15	15	0	5	10	10	0	5	5	5	0	1	2	2
19	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2	0	0
20	0	0	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2
21	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
22	0	10	20	20	0	8	15	15	0	5	10	10	0	5	5	5	0	1	2	2
23	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2	0	0
24	0	0	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2
25	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
26	0	10	20	20	0	8	15	15	0	5	10	10	0	3	5	5	0	1	2	2
27	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2	0	0
28	0	0	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2
29	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
30	0	10	20	20	0	8	15	15	0	5	10	10	0	3	5	5	0	1	2	2
31	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2	0	0
32	0	0	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2
33	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
34	0	10	20	20	0	8	15	15	0	5	10	10	0	3	5	5	0	1	2	2
35	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2	0	0
36	0	0	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2
37	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
38	0	10	20	20	0	8	15	15	0	5	10	10	0	3	5	5	0	1	2	2
39	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	2	2	0	0
40	0	0	10	10	0	0	8	8	0	0	5	5	0	0	2	2	0	0	1	1

__Table 37.__ Incoming of the Experiment 2.

d_{it}																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	5	5	5	5	3	3	3	3	2	2	2	2	0	0	0	0
2	0	0	0	0	5	5	5	5	3	3	3	3	1	1	1	1	0	0	0	0
3	5	5	5	5	5	5	5	5	3	3	3	3	2	2	2	2	0	0	0	0
4	5	5	5	5	5	5	5	5	3	3	3	3	1	1	1	1	0	0	0	0
5	5	5	5	5	5	5	5	5	3	3	3	3	2	2	2	2	0	0	0	0
6	5	5	5	5	5	5	5	5	3	3	3	3	1	1	1	1	0	0	0	0
7	5	5	5	5	5	5	5	5	3	3	3	3	2	2	2	2	0	0	0	0
8	5	5	5	5	5	5	5	5	3	3	3	3	1	1	1	1	0	0	0	0
9	5	5	5	5	5	5	5	5	3	3	3	3	2	2	2	2	0	0	0	0
10	5	5	5	5	5	5	5	5	3	3	3	3	1	1	1	1	0	0	0	0
11	13	13	13	13	11	11	11	11	6	6	6	6	4	4	4	4	2	2	2	2
12	13	13	13	13	8	8	8	8	6	6	6	6	3	3	3	3	2	2	2	2
13	13	13	13	13	11	11	11	11	6	6	6	6	4	4	4	4	2	2	2	2
14	13	13	13	13	8	8	8	8	6	6	6	6	3	3	3	3	2	2	2	2
15	13	13	13	13	11	11	11	11	6	6	6	6	4	4	4	4	2	2	2	2
16	13	13	13	13	8	8	8	8	6	6	6	6	3	3	3	3	2	2	2	2
17	13	13	13	13	11	11	11	11	6	6	6	6	4	4	4	4	2	2	2	2
18	13	13	13	13	8	8	8	8	6	6	6	6	3	3	3	3	2	2	2	2
19	13	13	13	13	11	11	11	11	6	6	6	6	4	4	4	4	2	2	2	2
20	13	13	13	13	7	7	7	7	6	6	6	6	3	3	3	3	2	2	2	2
21	12	12	12	12	9	9	9	9	6	6	6	6	3	3	3	3	1	1	1	1
22	12	12	12	12	7	7	7	7	6	6	6	6	2	2	2	2	1	1	1	1
23	12	12	12	12	9	9	9	9	6	6	6	6	3	3	3	3	1	1	1	1
24	12	12	12	12	7	7	7	7	6	6	6	6	2	2	2	2	1	1	1	1
25	12	12	12	12	9	9	9	9	6	6	6	6	3	3	3	3	1	1	1	1
26	12	12	12	12	7	7	7	7	6	6	6	6	2	2	2	2	1	1	1	1
27	12	12	12	12	9	9	9	9	6	6	6	6	3	3	3	3	1	1	1	1
28	12	12	12	12	7	7	7	7	6	6	6	6	2	2	2	2	1	1	1	1
29	12	12	12	12	9	9	9	9	6	6	6	6	3	3	3	3	1	1	1	1
30	12	12	12	12	7	7	7	7	6	6	6	6	2	2	2	2	1	1	1	1
31	10	10	10	10	7	7	7	7	5	5	5	5	3	3	3	3	1	1	1	1
32	10	10	10	10	8	8	8	8	5	5	5	5	2	2	2	2	1	1	1	1
33	10	10	10	10	7	7	7	7	5	5	5	5	3	3	3	3	1	1	1	1
34	10	10	10	10	8	8	8	8	5	5	5	5	2	2	2	2	1	1	1	1
35	10	10	10	10	7	7	7	7	5	5	5	5	3	3	3	3	1	1	1	1
36	10	10	10	10	8	8	8	8	5	5	5	5	2	2	2	2	1	1	1	1
37	10	10	10	10	7	7	7	7	5	5	5	5	3	3	3	3	1	1	1	1
38	10	10	10	10	8	8	8	8	5	5	5	5	2	2	2	2	1	1	1	1
39	10	10	10	10	7	7	7	7	5	5	5	5	3	3	3	3	1	1	1	1
40	10	10	10	10	8	8	8	8	5	5	5	5	2	2	2	2	1	1	1	1

Table 38. *Demand of the Experiment 2.*

B.1.3. Incoming and demand of the experiment 3

	q_{it}																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	5	5	5	13	5	7	7	10	5	5	5	5	2	3	3	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	15	10	10	10	0	8	0	0	0	5	10	10	0	3	0	0	0	1	2	2
7	0	0	0	0	0	0	15	15	0	0	0	0	0	0	5	5	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	4	3	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	30	30	0	0	0	0	0	0	10	10	0	0	0	0	0	0	2	2
17	0	0	0	0	0	0	15	15	0	0	0	0	0	0	5	5	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	20	10	0	0	15	7	0	0	10	5	0	0	5	2	0	0	2	1	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	10	20	20	0	0	0	0	0	5	10	10	0	3	0	0	0	1	3	3
27	0	0	0	0	0	8	15	15	0	0	0	0	0	0	5	5	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	20	20	0	0	15	15	0	0	10	10	0	0	5	5	0	0	4	4	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	10	10	0	0	0	0	0	0	5	5	0	0	0	0	0	0	1	1
37	0	0	0	0	0	0	6	6	0	0	0	0	0	0	2	2	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 39. Incoming of the Experiment 3.

d_{it}																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	1	1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
12	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
13	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
14	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
15	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	1	1	1	1
16	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
17	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
18	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
19	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
20	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
21	3	3	3	3	4	4	4	4	2	2	2	2	2	2	2	2	0	0	0	0
22	3	3	3	3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0
23	3	3	3	3	4	4	4	4	2	2	2	2	2	2	2	2	0	0	0	0
24	3	3	3	3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0
25	3	3	3	3	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2
26	3	3	3	3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0
27	3	3	3	3	4	4	4	4	2	2	2	2	2	2	2	2	0	0	0	0
28	3	3	3	3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0
29	3	3	3	3	4	4	4	4	2	2	2	2	2	2	2	2	0	0	0	0
30	3	3	3	3	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2
31	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
32	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
33	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
34	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
35	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	1	1	1	1
36	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
37	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
38	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
39	2	2	2	2	3	3	3	3	1	1	1	1	1	1	1	1	0	0	0	0
40	2	2	2	2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Table 40. *Demand of the Experiment 3.*

B.1.4. Incoming and demand of the experiment 4

	q_{it}																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	50	50	0	40	40	40	60	30	30	30	40	20	20	20	10	5	5	5
2	0	0	100	100	0	40	80	0	0	30	60	60	0	20	40	40	0	5	10	10
3	90	90	0	0	70	70	0	0	50	50	0	0	30	30	0	0	5	5	0	0
4	0	0	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10
5	100	50	0	0	80	40	0	0	60	30	0	0	40	20	0	0	10	5	0	0
6	0	50	100	100	0	40	80	95	0	30	60	60	0	20	40	40	0	8	10	10
7	110	110	0	0	90	80	0	0	60	60	0	0	40	40	0	0	13	10	0	0
8	0	0	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10
9	100	50	0	0	80	40	0	0	60	30	0	0	40	20	0	0	10	5	0	0
10	0	50	100	100	0	40	80	80	0	30	60	60	0	20	40	40	0	5	10	10
11	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10	0	0
12	0	0	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10
13	100	50	0	0	80	40	0	0	60	30	0	0	40	20	0	0	10	5	0	0
14	0	50	100	100	0	40	80	80	0	30	60	60	0	20	40	40	0	5	10	10
15	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10	0	0
16	0	0	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10
17	100	50	0	0	80	40	0	0	60	30	0	0	40	20	0	0	10	5	0	0
18	0	50	100	100	0	40	80	80	0	30	60	60	0	20	40	40	0	5	10	10
19	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10	0	0
20	0	0	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10
21	100	50	0	0	80	40	0	0	60	30	0	0	40	20	0	0	10	5	0	0
22	0	50	100	100	0	40	80	80	0	30	60	60	0	20	40	40	0	5	10	10
23	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10	0	0
24	0	0	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10
25	100	50	0	0	80	40	0	0	60	30	0	0	40	20	0	0	10	5	0	0
26	0	50	100	100	0	40	80	80	0	30	60	60	0	20	40	40	0	5	10	10
27	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10	0	0
28	0	0	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10
29	100	50	0	0	80	40	0	0	60	30	0	0	40	20	0	0	10	5	0	0
30	0	50	100	100	0	40	80	80	0	30	60	60	0	20	40	40	0	5	10	10
31	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10	0	0
32	0	0	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10
33	100	50	0	0	80	40	0	0	60	30	0	0	40	20	0	0	10	5	0	0
34	0	50	100	100	0	40	80	80	0	30	60	60	0	20	40	40	0	5	10	10
35	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10	0	0
36	0	0	100	100	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10
37	100	50	0	0	80	40	0	0	60	30	0	0	40	20	0	0	10	5	0	0
38	0	50	100	100	0	40	80	80	0	30	60	60	0	20	40	40	0	5	10	10
39	80	80	0	0	80	80	0	0	60	60	0	0	40	40	0	0	10	10	0	0
40	0	0	50	50	0	0	40	40	0	0	30	30	0	0	20	20	0	0	5	5

Table 41. *Incoming of the experiment 4.*

	d_{it}																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	40	40	0	30	30	0	20	20	20	20	10	10	10	10	2	2	2	2
2	0	0	40	40	0	30	30	0	20	20	20	20	10	10	10	10	2	2	2	2
3	40	40	40	40	30	30	30	30	20	20	20	20	10	10	10	10	2	2	2	2
4	40	40	40	40	30	30	30	30	20	20	20	20	10	10	10	10	2	2	2	2
5	40	40	40	40	30	30	30	30	20	20	20	20	10	10	10	10	2	2	2	2
6	40	40	40	40	30	30	30	30	20	20	20	20	10	10	10	10	2	2	2	2
7	40	40	40	40	30	30	30	30	20	20	20	20	10	10	10	10	2	2	2	2
8	40	40	40	40	30	30	30	30	20	20	20	20	10	10	10	10	2	2	2	2
9	40	40	40	40	30	30	30	30	20	20	20	20	10	10	10	10	2	2	2	2
10	40	40	40	40	30	30	30	30	20	20	20	20	10	10	10	10	2	2	2	2
11	50	50	50	50	40	40	40	40	30	30	30	30	20	20	20	20	5	5	5	5
12	50	50	50	50	40	40	40	40	30	30	30	30	20	20	20	20	5	5	5	5
13	60	60	60	60	50	50	50	50	40	40	40	40	30	30	30	30	8	8	8	8
14	50	50	50	50	40	40	40	40	30	30	30	30	20	20	20	20	5	5	5	5
15	50	50	50	50	40	40	40	40	30	30	30	30	20	20	20	20	5	5	5	5
16	40	40	50	50	30	30	40	40	20	20	30	30	10	10	20	20	0	0	5	5
17	50	50	50	50	40	40	40	40	30	30	30	30	20	20	20	20	5	5	5	5
18	50	50	50	50	40	40	40	40	30	30	30	30	20	20	20	20	5	5	5	5
19	50	50	50	50	40	40	40	40	30	30	30	30	20	20	20	20	5	5	5	5
20	50	50	50	50	40	40	40	40	30	30	30	30	20	20	20	20	5	5	5	5
21	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	8	8	8	8
22	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	8	8	8	8
23	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	8	8	8	8
24	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	8	8	8	8
25	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	8	8	8	8
26	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	8	8	8	8
27	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	8	8	8	8
28	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	8	8	8	8
29	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	8	8	8	8
30	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	8	8	8	8
31	50	50	50	50	40	40	40	40	30	30	30	30	20	20	20	20	2	2	2	2
32	50	50	50	50	40	40	40	40	30	30	30	30	20	20	20	20	5	5	5	5
33	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	5	5	5	5
34	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	5	5	5	5
35	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	5	5	5	5
36	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	5	5	5	5
37	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	5	5	5	5
38	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	5	5	5	5
39	55	55	55	55	45	45	45	45	35	35	35	35	25	25	25	25	5	5	5	5
40	0	0	55	55	0	45	45	0	35	35	35	35	25	25	25	25	5	5	5	5

Table 42. *Demand of the Experiment 4.*

B.2. Inventory level

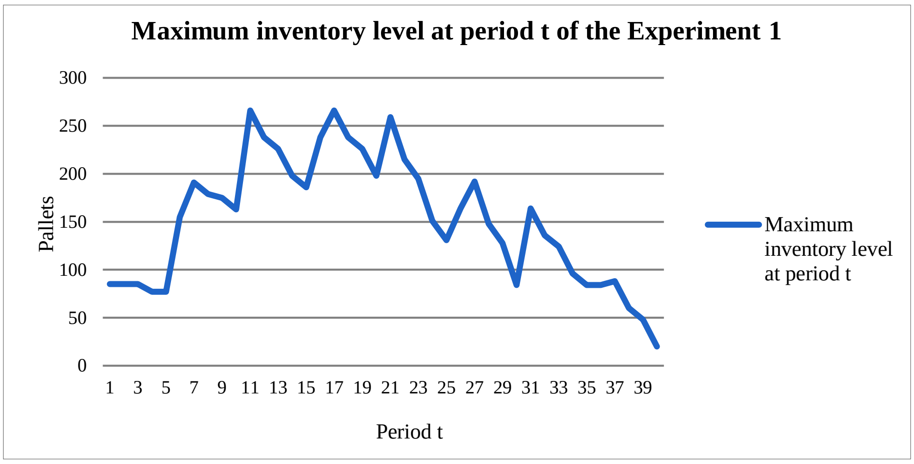


Figure 39. Maximum inventory level *of pallets at period t of the experiment with a low volume of stock and a low rotation of stock.*

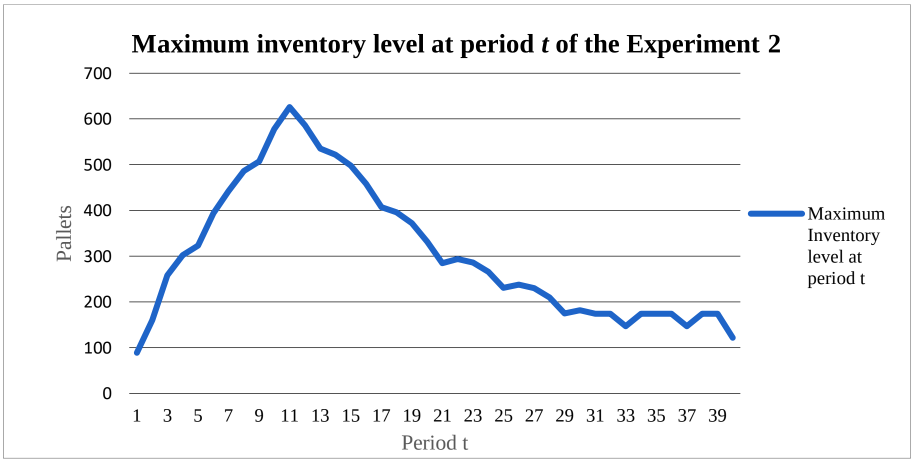


Figure 40. *Maximum inventory level of pallets at period t of the experiment with a low volume of stock and a high rotation of stock.*

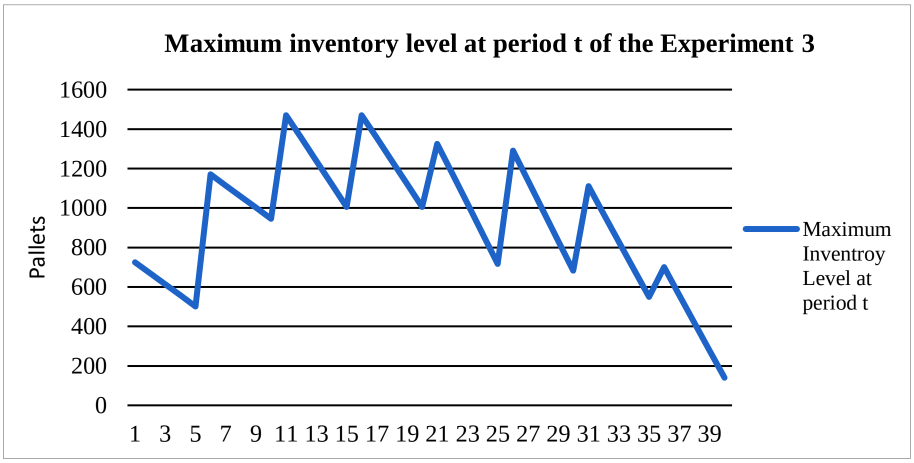


Figure 41. *Maximum inventory level of pallets at period t of the Experiment with a high volume of stock and a high rotation of stock.*

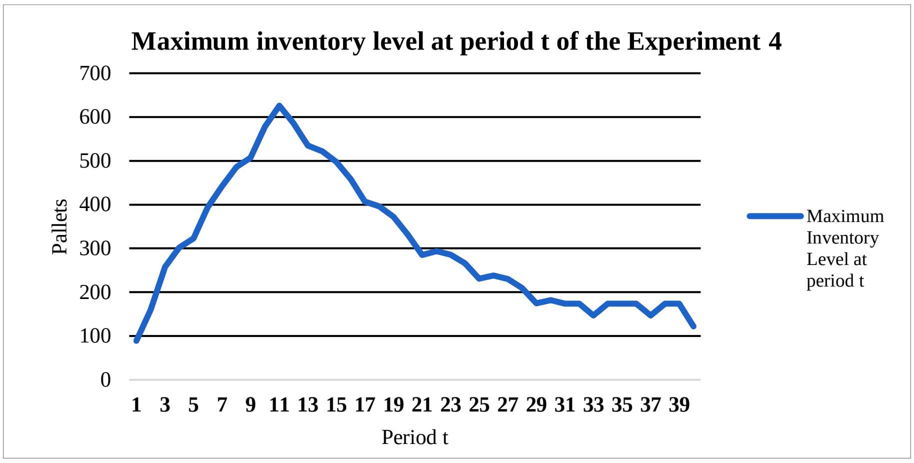


Figure 42. The maximum inventory level of pallets at period t of the experiment with a high volume of stock and a high rotation of stock.

B.3. Results of the maximum number of single-deep and multi-deep lanes after going through each process of the Model

	Results of the combination of single-deep and multi-deep lanes of experiment 1														
k	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Local Optimization	12	23	9	0	0	0	0	0	0	0	0	0	0	0	0
Relocation	20	19	9	0	0	0	0	0	0	0	0	0	0	0	0
Redistribution (Initial Peak Redistribution)	20	13	8	0	0	0	0	0	0	0	0	0	0	0	0
Redistribution (Backward Redistribution)	24	11	7	0	0	0	0	0	0	0	0	0	0	0	0
Redistribution (Forward Redistribution)	24	11	7	0	0	0	0	0	0	0	0	0	0	0	0

Table 43. Results of the combination of single-deep and multi-deep lanes of different depth of the experiment with a low volume of stock and a low rotation.

	Results of the combination of single-deep and multi-deep lanes of experiment 2														
k	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Local Optimization	35	63	14	2	0	0	0	0	0	0	0	0	0	0	0
Relocation	42	55	13	2	0	0	0	0	0	0	0	0	0	0	0
Redistribution (Initial Peak Redistribution)	39	59	8	0	0	0	0	0	0	0	0	0	0	0	0
Redistribution (Backward Redistribution)	38	53	5	0	0	0	0	0	0	0	0	0	0	0	0
Redistribution (Forward Redistribution)	38	53	5	0	0	0	0	0	0	0	0	0	0	0	0

Table 44. *Results of the combination of single-deep and multi-deep lanes of different depth of the experiment with a low volume of stock and a high rotation.*

	Results of the combination of single-deep and multi-deep lanes of experiment 3														
k	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Local Optimization	4	26	20	30	22	21	0	0	0	0	0	0	0	0	0
Relocation	13	17	0	31	40	0	0	0	0	0	0	0	0	0	0
Redistribution (Initial Peak Redistribution)	16	22	0	32	28	0	0	0	0	0	0	0	0	0	0
Redistribution (Backward Redistribution)	16	26	0	27	30	0	0	0	0	0	0	0	0	0	0
Redistribution (Forward Redistribution)	16	26	0	27	30	0	0	0	0	0	0	0	0	0	0

Table 45. *Results of the combination of single-deep and multi-deep lanes of different depth of the experiment with a high volume of stock and a low rotation.*

	Results of the combination of single-deep and multi-deep lanes of experiment 4														
k	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Local Optimization	66	51	25	29	21	20	0	0	0	0	0	0	0	0	28
Relocation	69	128	0	90	0	0	0	0	0	0	0	0	0	0	7
Redistribution (Initial Peak Redistribution)	61	117	0	57	0	0	0	0	0	0	0	0	0	0	6
Redistribution (Backward Redistribution)	65	155	0	37	0	0	0	0	0	0	0	0	0	0	1
Redistribution (Forward Redistribution)	65	155	0	37	0	0	0	0	0	0	0	0	0	0	1

Table 46. *Results of the combination of single-deep and multi-deep lanes of different depth of the experiment with a high volume of stock and a high rotation.*